

Sistema de community cloud aplicat a la  
preservació digital personal (personal archiving)  
PYRAMID

Carlos García Rubio

## Agraïments

M'agradaria agrair i dedicar aquest projecte a totes les persones que han fet possible la realització d'aquest projecte.

Al Dr. Josep Lluís de la Rosa per donar-me la oportunitat d'entrar al projecte Pyramid i formar part de l'equip del Centre Easy, tanmateix, donar les gràcies al meu tutor Albert Bres Segalés per tutoritzar-me i ajudar-me en la realització d'aquest.

Al doctorant Jose Antonio Olvera Cañizares per ajudar-me amb la teoria de la preservació digital i el sistema de disseny d'agents INGENIAS.

A la meua xicota i a les persones que m'han recolzat dia a dia per tal de portar a bon terme la realització d'aquest projecte i la documentació.

A tots vosaltres, gràcies.

# Índex

<b>1</b>	<b>Introducció</b>	<b>14</b>
1.1	Motivació i propòsits . . . . .	14
1.2	Objectius . . . . .	15
1.3	Antecedents . . . . .	17
1.4	Estudi de viabilitat . . . . .	17
1.5	Metodologia . . . . .	18
1.5.1	Iconix . . . . .	18
1.5.2	INGENIAS . . . . .	19
1.6	Planificació . . . . .	22
1.6.1	Diagrama de Gantt . . . . .	22
1.7	Estructura de la documentació . . . . .	24
<b>2</b>	<b>Conceptes previs</b>	<b>25</b>
2.1	Intel·ligència artificial . . . . .	25
2.1.1	El concepte d' <i>Agent intel·ligent</i> . . . . .	26
2.1.1.1	Els agents com a metàfora de disseny . . . . .	28
2.1.1.2	Els agents com a simulació . . . . .	28
2.1.2	Sistema multiagent . . . . .	29
2.2	Preservació Digital (PD) . . . . .	30
2.2.1	Què és la PD? . . . . .	30
2.2.2	Estratègies de PD . . . . .	31
2.2.2.1	Preservació de la tecnologia . . . . .	31

ÍNDIX	3
2.2.2.2 Emulació de la tecnologia . . . . .	32
2.2.2.3 Migració de la informació . . . . .	32
2.2.2.4 Encapsulació . . . . .	33
2.3 Swarm intelligence . . . . .	33
2.3.1 La colònia de les formigues . . . . .	34
2.4 Computational ecologies . . . . .	36
2.4.1 Algoritmes evolutius i algoritmes genètics . . . . .	37
2.4.1.1 Algoritmes genètics . . . . .	37
2.5 Community Cloud . . . . .	40
2.5.1 Cloud Computing . . . . .	40
2.5.2 Community Cloud Computing . . . . .	40
2.6 Final del capítol . . . . .	41
<b>3 Requeriments del sistema</b>	<b>42</b>
3.1 Requeriments funcionals . . . . .	42
3.2 Requeriments no funcionals . . . . .	42
3.3 Estudis i decisions . . . . .	43
3.3.1 Base de Dades (BD) i Servidor . . . . .	43
3.4 Software . . . . .	45
<b>4 Anàlisi</b>	<b>50</b>
4.1 Prototipus de GUI . . . . .	50
4.1.1 Login . . . . .	50
4.1.2 Pantalla principal . . . . .	51
4.1.3 Administració de patrimoni . . . . .	52
4.1.4 Afegir col·lecció . . . . .	53
4.1.5 Pantalla de configuració . . . . .	53
4.1.6 Pantalla d'actualitzacions . . . . .	54
4.1.7 Pantalla de <i>plugins</i> . . . . .	54
4.1.8 <i>Tray icon</i> . . . . .	55

4.1.9	Pantalla d'amics . . . . .	55
4.1.10	Pantalla d' <i>about</i> . . . . .	56
4.2	Casos d'us . . . . .	57
4.2.1	Iniciar sessió . . . . .	57
4.2.2	Menú principal . . . . .	58
4.2.3	Gestionar col·leccions . . . . .	59
4.3	Fitxes de cas d'us . . . . .	60
4.3.1	Iniciar sessió . . . . .	60
4.3.2	Crear col·lecció . . . . .	60
4.3.3	Eliminar col·lecció . . . . .	61
4.4	Diagrames de seqüència . . . . .	61
4.4.1	Iniciar sessió . . . . .	61
4.4.2	Crear col·lecció . . . . .	63
4.5	Diagrama de robustesa . . . . .	64
4.6	Diagrama de classes . . . . .	65
4.6.1	Disseny en tres capes . . . . .	65
4.6.2	Capes . . . . .	67
4.6.2.1	Mòdul Pyramid . . . . .	67
4.6.2.2	Mòdul PyramidBusiness . . . . .	67
4.6.2.3	Mòdul PyramidDataLocal . . . . .	67
4.7	Anàlisi INGENIAS . . . . .	71
4.7.1	Model d'organització . . . . .	71
4.7.2	Models d'Agents . . . . .	72
4.7.2.1	Agent Preservador . . . . .	72
4.7.2.2	Agent Recomanador . . . . .	73
4.7.3	Models d'interacció . . . . .	74
4.7.3.1	Interacció Preservador-Recomanador . . . . .	74
4.7.3.2	Interacció Recomanador Local-Recomanador Remot . . . . .	75

<i>ÍNDEX</i>	5
5.1 Disseny en tres capes . . . . .	76
5.2 Observer . . . . .	78
5.3 Visitor . . . . .	79
5.4 Factory Method . . . . .	80
5.5 Singleton . . . . .	81
<b>6 Implementació i Proves</b>	<b>82</b>
6.1 Problemes . . . . .	82
6.2 Preservadors . . . . .	82
6.2.1 Interfície . . . . .	82
6.2.2 Instanciar preservador . . . . .	84
6.2.3 Ampliació de preservadors . . . . .	86
6.2.3.1 Exemple . . . . .	86
6.3 Preservar . . . . .	89
6.3.1 Afegir col·lecció . . . . .	89
6.3.2 Fil principal de preservació . . . . .	90
6.3.2.1 Destins de preservació . . . . .	91
6.3.2.2 Mètodes de preservació . . . . .	94
6.3.3 Restaurar . . . . .	96
6.3.3.1 Versió usuari . . . . .	96
6.3.3.2 Versió programador . . . . .	98
6.4 Mòdul multiidioma . . . . .	99
6.4.1 Implementació Visitor . . . . .	100
6.4.2 Com escollir idioma . . . . .	101
6.5 Log . . . . .	102
6.6 Proves . . . . .	103
6.6.1 Autenticar l'usuari . . . . .	103
6.6.1.1 Login erroni . . . . .	103
6.6.1.2 Login correcte . . . . .	104
6.6.2 Administrar àlbums . . . . .	105

6.6.2.1	Crear àlbum . . . . .	105
6.6.2.2	Modificar àlbum . . . . .	106
6.6.2.3	Vista en miniatura . . . . .	107
6.6.2.4	Eliminar àlbum . . . . .	108
6.6.2.5	Confirmació visual de canvi . . . . .	109
6.6.3	Preservació i restauració . . . . .	110
6.6.3.1	Preservacions . . . . .	111
6.6.3.2	Restaurar . . . . .	112
6.6.4	Amics i formats de moda . . . . .	113
6.6.4.1	Formats de moda . . . . .	114
6.6.5	Multiidioma . . . . .	115
6.6.5.1	Català . . . . .	115
6.6.5.2	Anglès . . . . .	116
6.6.6	Ampliació dinàmica de preservadors . . . . .	117
6.6.7	Execució en <i>Background</i> . . . . .	118
<b>7</b>	<b>Conclusions</b>	<b>119</b>
7.1	Avaluació dels objectius . . . . .	119
7.2	Avaluació dels requeriments . . . . .	120
7.2.1	Requeriments funcionals . . . . .	120
7.2.2	Requeriments no funcionals . . . . .	120
7.3	Problemes trobats . . . . .	121
7.4	Treball futur . . . . .	122
<b>8</b>	<b>Manual d'usuari</b>	<b>123</b>
8.1	Descàrrega i instal·lació . . . . .	123
8.2	Utilització . . . . .	125
8.2.1	Pantalla <i>Login</i> . . . . .	125
8.2.1.1	Registrar . . . . .	126
8.2.2	Pantalla Principal . . . . .	127

8.2.3	Pantalla opcions . . . . .	127
8.2.3.1	Perfil . . . . .	128
8.2.3.2	Preferències . . . . .	128
8.2.3.3	Àlbums . . . . .	129
8.2.3.4	Amics . . . . .	130
<b>A</b>	<b>It is Forever</b>	<b>132</b>
A.1	Introducció . . . . .	132
A.2	Què és? . . . . .	133
A.3	Unió a la inversa . . . . .	133
<b>B</b>	<b>Problemes en el Desenvolupament Basat en Agents</b>	<b>136</b>



# Índex de figures

1.1	Diagrama Pyramid . . . . .	16
1.2	Logo del projecte PROTAGE. . . . .	17
1.3	Diagrama Iconix. . . . .	18
1.4	Logo Iconix. . . . .	19
1.5	Relacions entre els diferents meta-models d'INGENIAS i les dues entitats principals: la organització i l'agent. . . . .	21
1.6	Diagrama de seqüència d'iniciar sessió. . . . .	23
2.1	Visió esquemàtica d'un agent intel·ligent . . . . .	27
2.2	Analogia entre la preservació digital i la preservació de les formigues. . . . .	35
2.3	Individu genètic binari. . . . .	39
2.4	Cloud Computing . . . . .	40
2.5	Community Cloud . . . . .	41
3.1	Diagrama entitat-relació. . . . .	44
3.2	Visual Studio 2010 Professional. . . . .	45
3.3	Logo de MySQL. . . . .	46
3.4	Visual Basic. . . . .	46
3.5	Logo de .NET. . . . .	46
3.6	Software per a la creació de UML . . . . .	47
3.7	Logo de Gantt Project. . . . .	47
3.8	Logo de Gliffy. . . . .	48
3.9	Logo de Source Safe 2005. . . . .	48

3.10	Logo de XAML. . . . .	49
3.11	Logo de FileZilla. . . . .	49
4.1	Pantalla de Login. . . . .	51
4.2	Pantalla principal. . . . .	51
4.3	Administració de patrimoni. . . . .	52
4.4	Afegir col·lecció. . . . .	53
4.5	Pantalla de configuració. . . . .	53
4.6	Pantalla de les actualitzacions. . . . .	54
4.7	Pantalla dels plugins. . . . .	54
4.8	Tray icon. . . . .	55
4.9	Pantalla de la gestió d'amics. . . . .	55
4.10	Pantalla d'about. . . . .	56
4.11	Cas d'ús de Inici de sessió. . . . .	57
4.12	Menú principal de l'aplicació. . . . .	58
4.13	Diagrama de cas d'ús de la gestió de col·leccions. . . . .	59
4.14	Diagrama de seqüència d'iniciar sessió. . . . .	62
4.15	Diagrama de seqüència de crear col·lecció. . . . .	63
4.16	Diagrama de seqüència de crear col·lecció. . . . .	64
4.17	Diagrama de classes 3 capes . . . . .	66
4.18	Diagrama de classes Pyramid . . . . .	68
4.19	Diagrama de classes PyramidBusiness . . . . .	69
4.20	Diagrama de classes PyramidDataLocal . . . . .	70
4.21	Model d'organització . . . . .	71
4.22	Model d'agent Preservador . . . . .	72
4.23	Model d'agent Recomanador . . . . .	73
4.24	Model interacció Preservador-Recomanador . . . . .	74
4.25	Model interacció Recomanador Local-Recomanador Remot . . . . .	75
5.1	Diagrama d'arquitectura en tres capes. . . . .	76
5.2	Imatge de la jerarquia de capes . . . . .	77

5.3	Diagrama Observer. . . . .	78
5.4	Diagrama Visitor. . . . .	79
5.5	Diagrama Factory Method. . . . .	80
5.6	Diagrama Singleton. . . . .	81
6.1	Interfície del Preservador . . . . .	84
6.2	Diccionaris dels preservadors locals . . . . .	85
6.3	Codi del creador de preservadors . . . . .	85
6.4	Descàrrega del preservador . . . . .	86
6.5	Nou projecte llibreria . . . . .	86
6.6	Referències . . . . .	86
6.7	Nou preservador . . . . .	87
6.8	Dll penjada al servidor . . . . .	88
6.9	Afegir Col·lecció . . . . .	89
6.10	Codi per crear nova Col·lecció . . . . .	89
6.11	Imatge preservada . . . . .	89
6.12	Thread principal . . . . .	90
6.13	Funció Preservar . . . . .	91
6.14	Preservació local . . . . .	92
6.15	Preservació remota . . . . .	93
6.16	Preservació amiga . . . . .	93
6.17	Preservació copia . . . . .	94
6.18	Preservació transformació . . . . .	95
6.19	Funció convertir . . . . .	95
6.20	Funció convertirBytes . . . . .	96
6.21	Restaurar arxiu . . . . .	97
6.22	Imatge restaurada . . . . .	97
6.23	Part comuna de restauració . . . . .	98
6.24	Restaurar remot . . . . .	98
6.25	Restaurar local . . . . .	99

6.26 Restaurar amic . . . . .	99
6.27 Llista de recursos . . . . .	100
6.28 Missatges en català . . . . .	100
6.29 Escollir missatge . . . . .	100
6.30 Escollir idioma . . . . .	101
6.31 Codi d'escollir idioma . . . . .	101
6.32 Classe Log . . . . .	102
6.33 Login erroni . . . . .	103
6.34 Login error . . . . .	103
6.35 Login correcta . . . . .	104
6.36 Pantalla Principal . . . . .	104
6.37 Nou album . . . . .	105
6.38 Guardar album . . . . .	106
6.39 Vista en miniatura . . . . .	107
6.40 Eliminar àlbum . . . . .	108
6.41 Album Eliminat . . . . .	108
6.42 Nou Element . . . . .	109
6.43 Diferents preservacions . . . . .	110
6.44 Preservació remota . . . . .	111
6.45 Preservació local . . . . .	111
6.46 Petició de restauració . . . . .	112
6.47 Restaurat . . . . .	112
6.48 Afegir amic . . . . .	113
6.49 Petició amic . . . . .	113
6.50 Amics i formats . . . . .	114
6.51 Aplicació en català . . . . .	115
6.52 Aplicació en angles . . . . .	116
6.53 DLL al servidor . . . . .	117
6.54 Imatge jpg transformada a altres formats . . . . .	117

6.55	Transformacions al servidor . . . . .	118
6.56	Aplicació en segon pla . . . . .	118
6.57	Consum de l'aplicació . . . . .	118
7.1	Diagrama actualment . . . . .	122
7.2	Diagrama en el futur . . . . .	122
8.1	Web pyramid . . . . .	123
8.2	Instal·lador . . . . .	124
8.3	Instal·lació correcte . . . . .	124
8.4	Login . . . . .	125
8.5	nou password . . . . .	125
8.6	Nou Usuari . . . . .	126
8.7	Correu de validació . . . . .	126
8.8	Pantalla principal . . . . .	127
8.9	Pantalla perfil . . . . .	128
8.10	Pantalla preferencies . . . . .	129
8.11	Pantalla albums . . . . .	130
8.12	Pantalla amics . . . . .	131
A.1	It Is Forever Logo . . . . .	132
A.2	Nou ItIsForever . . . . .	133
A.3	It Is Forever Àlbum . . . . .	134
A.4	Entitat relació final . . . . .	135

# Índex de taules

4.1	Fitxa de cas d'us d'iniciar sessió . . . . .	60
4.2	Fitxa de cas d'us de crear col·lecció . . . . .	60
4.3	Fitxa de cas d'us d'eliminar col·lecció . . . . .	61

# Capítol 1

## Introducció

### 1.1 Motivació i propòsits

La preservació digital (PD) s'ha convertit en un problema persistent per a tots els que vulguin conservar la seva informació digital, garantir el seu estat i consultar aquest informació en el transcurs del temps. Fins ara només grans institucions amb coneixement expert i eines especialitzades han pogut fer front a aquest problema, però la preservació digital no pot ser abordada per una sola institució o nació. Les biblioteques, arxius i altres institucions de conservació de la memòria comparteixen aquest repte al igual que amb els col·leccionistes i creadors que el fan a títol individual.

A l'actualitat, el nivell d'automatització en les solucions de la conservació digital és pobre, mentre que els volums d'informació, la diversitat de formats i tipus d'objectes digitals van augmentant de forma exponencial. El procés de conservació a l'actualitat té moltes etapes manuals, i ha d'abordar-se de manera flexible i distribuïda, mitjançant la combinació de mètodes automàtics intel·ligents amb la intervenció humana. L'escalabilitat de les solucions de preservació existents ha mostrat ser limitada. A més, les solucions no s'han provat adequadament amb diversos recursos digitals o en entorns heterogènic. Sembla que la investigació en PD s'ha allunyat de l'objectiu de trobar una única solució ideal i s'ha centrat en la definició de solucions pràctiques per a diferents situacions de preservació. Aquestes solucions haurien d'aprofitar els coneixements d'experts d'institucions de memòria basant-se en estàndards de la indústria i sobretot, ser escalable i adaptable a diferents entorns.

Aprofitant que hi ha una recerca del grup ARLab en la línia d'agents i preservació digital ([44], [47], [46]), intel·ligència computacional [45], ecosistemes digitals[51], i altres temes relacionats que recomana implementar unes primitives del comporta-

ment dels objectes digitals[52] (migrar d'una determinada manera ja sigui en còpia o en mutació), es precisa d'una aplicació en versió alpha que implementi aquestes idees.

Així doncs, el propòsit d'aquest projecte és crear l'aplicació Pyramid que està concebuda com una eina de suport orientada a l'usuari domèstic [43] (sense coneixements tècnics ni de preservació) per a la preservació a mig i llarg termini de col·leccions digitals, texts i vídeos, tal que funcioni com un antivirus (en *BackGround*) i preservi la informació sense requerir un cost addicional a l'ordinador i que l'usuari no noti cap molèstia a l'hora de fer les seves tasques diàries.

## 1.2 Objectius

L'objectiu final és obtenir una aplicació per a l'usuari domèstic per a que pugui preservar els seus objectes digitals.

Per a poder arribar a aquest fi últim, caldrà un estudi previ dels següents punts:

- Aprendre les bases teòriques de la preservació digital.
- Estudi dels algoritmes de *swarm intelligence* i *computational ecologies* i la seva aplicació pràctica a la PD.
- Estudi de diferents llenguatges de programació i les diferents llibreries per facilitar la preservació.

Un cop assolits els punts anteriors, l'objectiu final al que vol arribar aquest projecte està dividit en els següents punts:

- Implementació d'una aplicació per a la PD d'us domèstic.
- Aprofitar el motor d'aquesta aplicació per a la implementació d'altres serveis de PD.
- Permetre, gràcies a l'estudi i l'elaboració d'aquest projecte, continuar investigant sobre la PD i/o tasques relacionades amb aquesta.



Un esquema del recorregut que han de fer els arxius a través de l'aplicació podria ser el següent.

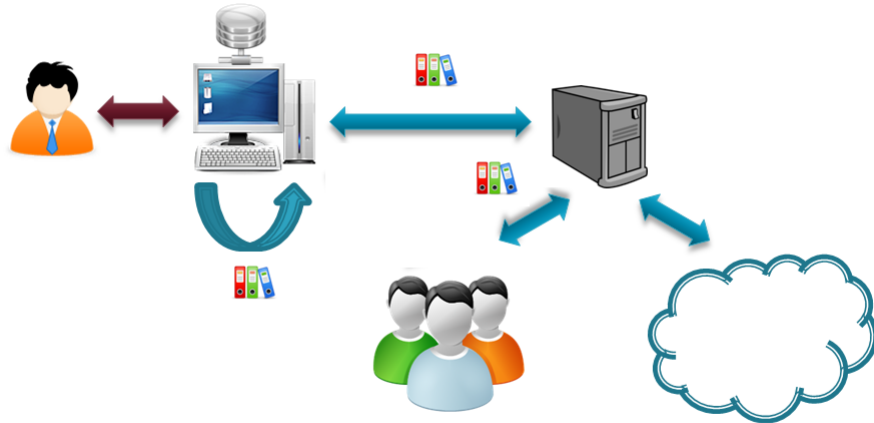


Figura 1.1: Recorregut dels arxius

## 1.3 Antecedents

La idea de Pyramid va sorgir a través d'un projecte anomenat PROTAGE<sup>1</sup>[2] [48] [49] amb la idea de portar aquest, que llavors estava pensat per a institucions, cap a l'usuari domèstic [43] [50].



Figura 1.2: Logo del projecte PROTAGE.

## 1.4 Estudi de viabilitat

Aquest projecte forma part de la investigació que es duu a terme en el grup de recerca ARLab (Agents Research Laboratory) dins el departament EEEA (Enginyeria Elèctrica, Electrònica i Automàtica).

Per a la realització del projecte, necessitem els següents elements:

- Un ordinador
- Base de dades MySQL montada en un servidor
- L'IDE ( *Integrated Development Environment*) Microsoft Visual Studio
- Llibreries .Net
- Programari L<sup>A</sup>T<sub>E</sub>X per a la redacció de la memòria.
- Un gestor de tasques per al seguiment dels objectius i feina pendent anomenat *ClockingIT*[4]

Els elements de software són de lliure distribució o sota llicències d'estudiants i els elements de Hardware els aporta el grup de recerca, per tan, el projecte és viable.

---

<sup>1</sup>(*PReservation Organizations using Tools in AGent Environments*) desenvolupat per a investigar el potencial dels agents per recolzar les tasques de preservació i explorar/analitzar la integració en diferents aplicatius de PD i la comercialització d'aquests.

## 1.5 Metodologia

Per a la metodologia, hem de separar la part de l'aplicació, és a dir, l'interfície d'usuari i la plataforma per una banda i el paradigma d'agents per una altra.

Per a la primera part, farem servir una metodologia híbrida denominada *Iconix*[3] i per a la part d'agents, una metodologia anomenada *INGENIAS*[39].

S'ha escollit aquesta separació degut a que la interfície d'usuari no és agentificable, no es pot agentificar tot [41].

### 1.5.1 Iconix

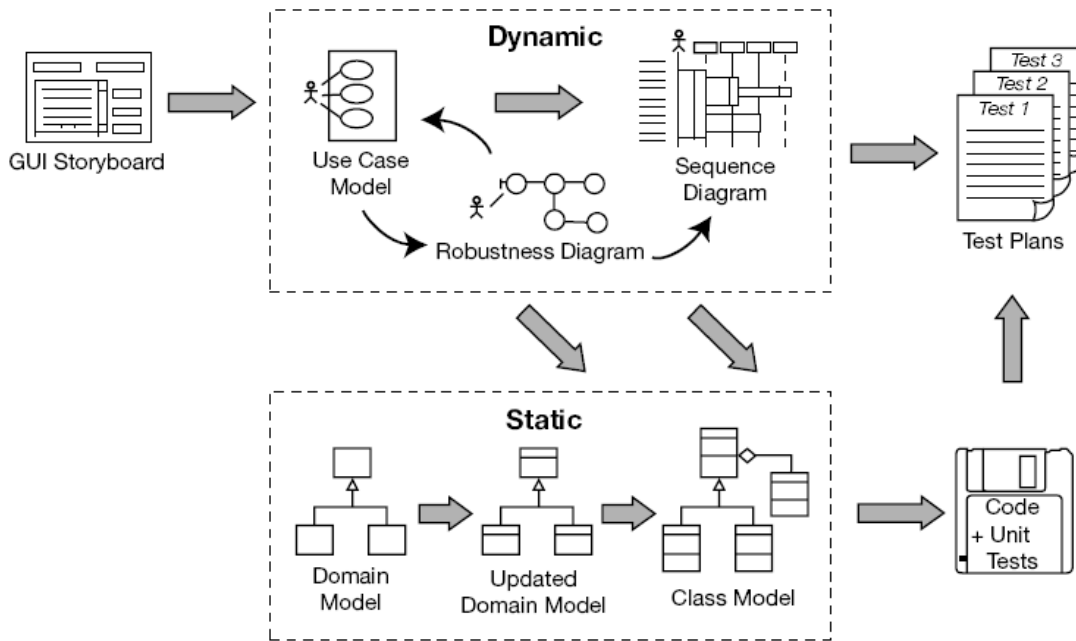


Figura 1.3: Diagrama Iconix.

La metodologia *Iconix*[3] és una metodologia que, sense acabar d'encaixar en la llista de metodologies àgils ni la de metodologies pesades, sí adopta elements d'ambdós tipus.

Per una banda, adopta elements de la metodologia pesada "RUP" (*Rational Unified Process*) i elements de la metodologia àgil "XP" (*Extreme Programming*).

*Iconix* deriva directament de la metodologia RUP i argumenta que en el 80% de les ocasions, els projectes es poden realitzar amb èxit utilitzant només un 20% de UML,



Figura 1.4: Logo Iconix.

simplificant molt el procés però sense perdre documentació. Aquesta metodologia es guia a través dels casos d'ús i segueix un cicle de vida iteratiu i incremental. L'objectiu és que a partir dels casos d'ús s'obtingui el resultat final. La metodologia Iconix s'estructura en quatre fases diferenciades:

- Anàlisi de requeriments
- Anàlisi i disseny preliminar
- Disseny
- Implementació

El pas previ a aquestes quatre fases és traçar un esbós o storyboard de les interfícies gràfiques de l'aplicació.

La qualitat del sistema s'avalua comprovant si s'han assolit els objectius i s'han complert els requeriments especificats en la fase prèvia a l'anàlisi.

En aquest projecte, el tutor realitza el rol de client i serà amb ell amb qui s'analitzaran els requeriments i es validaran els mòduls que es vagin acabant d'implementar.

## 1.5.2 INGENIAS

Els agents tenen les seves pròpies metodologies, entre elles tenim *Jade*, *ABLE*, *ZEUS*, *MESSAGE*, etc. Nosaltres hem escollit la metodologia que evoluciona d'aquesta última: *INGENIAS*[39].

S'ha escollit aquesta metodologia, entre altres raons, degut a que es va emprar en el projecte anterior a aquest anomenat *PROTAGE*

Hi ha moltes aproximacions vàlides per al desenvolupament de Sistemes Multi-Agent (SMA) basades en diferents idees de què hauria de ser un Sistema Multi-Agent. Un intent cap a una integració de metodologies orientades a agents va ser *MESSAGE* [40]. La principal contribució de *MESSAGE* va ser la definició de meta-models<sup>2</sup> per a l'especificació d'elements que poden ser emprats per descriure cadascun dels aspectes

---

<sup>2</sup>Un meta-model és una representació dels tipus d'entitats que poden existir en un model, les seves relacions i restriccions d'aplicació. Com ha demostrat UML, construït també amb meta-models, aquest tipus de notació facilita enormement el desenvolupament de sistemes.

que constitueixen un sistema multi-agent des de cinc punts de vista: organització, agents, objectius/tasques, interaccions i domini. MESSAGE va adoptar Unified Process (UP) i es va centrar en les fases de desenvolupament de l'anàlisi i disseny.

*INGENIAS* parteix dels resultats de MESSAGE i proporciona meta-models més complets i consistents, basats en més experiència en la construcció d'aplicacions basades en agents, i un conjunt d'eines que donen suport a l'anàlisi, disseny i activitats de generació de codi.

Cadascun dels meta-models mostra una visió parcial del SMA: els agents que el componen, les interaccions que existeixen entre ells, com s'organitzen per proporcionar la funcionalitat del sistema, quina informació és rellevant en el domini i com és l'entorn en el que s'ubica el sistema a desenvolupar. A continuació es descriuen breument:

- **Meta-model d'organització:** és l'equivalent a l'arquitectura del sistema. Descriu com s'agrupen i coordinen els diferents components del sistema, identifica les tasques rellevants per l'organització, així com els seus objectius globals, i defineix restriccions en les interaccions entre els agents.
- **Meta-model d'agent:** descriu agents particulars, les seves tasques, objectius, els rols que duen a terme i els estats mentals en que es trobaran al llarg de la seva vida.
- **Meta-model d'objectius i tasques:** té com a propòsit recollir motivacions del sistema d'agents, definir les relacions entre objectius i tasques, indicar les entrades i sortides de cada tasca, i descriure quins són els seus efectes, tant en l'entorn com en l'estat mental dels seus agents responsables.
- **Meta-model d'interacció:** descriu com es coordinen i comuniquen els agents del sistema. Així inclou als actors implicats, les unitats d'interacció, els protocols emprats i com afecta la interacció en el context.
- **Meta-model d'entorn:** En aquest meta-model el propòsit no és generar representacions del món en el que s'ubica el sistema, sinó, des d'un punt de vista més pragmàtic, categoritzar el tipus d'entitats rellevants en l'entorn i restringir la interacció amb aquestes. Això, l'entorn contindrà només recursos, aplicacions i agents i es limitarà la percepció i actuació dels agents.

Aquests cinc meta-models giren al voltant de dues entitats: l'organització i l'agent.

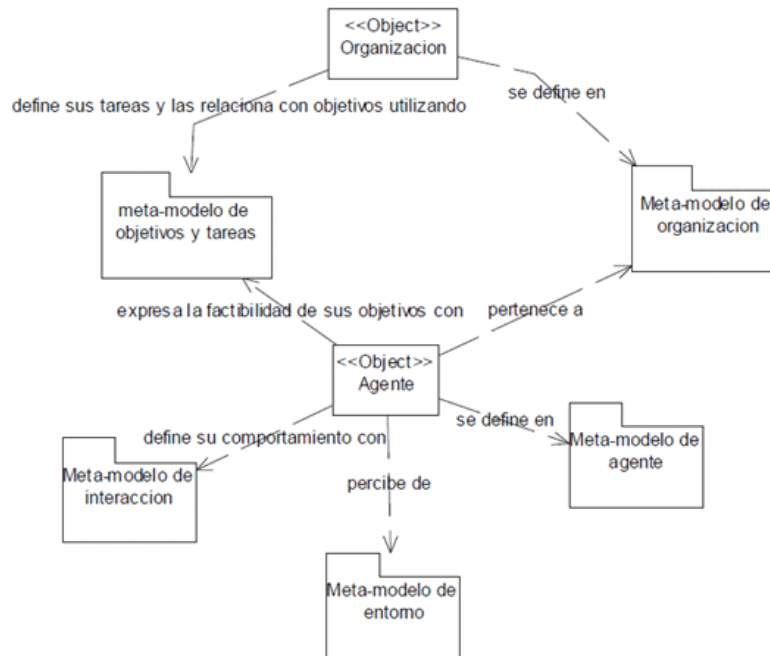


Figura 1.5: Relacions entre els diferents meta-models d'INGENIAS i les dues entitats principals: la organització i l'agent.

INGENIAS també proporciona eines gràfiques per generar especificacions, validar els dissenys, generar codi de forma automàtica i generar documents, entre altres coses. El nom del software és INGENIAS Development Kit (IDK).

El procés de desenvolupament del cicle de vida adopta l'Unified Process, definint les activitats que s'han de dur a terme en diferents etapes del desenvolupament del cicle de vida. La integració en el cicle de vida s'aconsegueix definint un conjunt d'entregues i activitats involucrades en el desenvolupament.

Finalment, cal esmentar també que aquesta metodologia no està completa en el sentit de que el llenguatge d'especificació de SMA que proposa descansa sobre resultats d'investigacions que creixen dia a dia.

## 1.6 Planificació

- **Estudi i planificació:** El primer apartat es basa en fer un estudi dels requeriments de l'aplicatiu, a conèixer, les taules que hi haurà a la base de dades, quin tipus de base de dades fer servir, els requeriments (tan funcionals com no funcionals) inicials, segurament se n'afegiran en el transcurs de la implementació (recordem que Iconix és una metodologia iterativa), prototipus de la GUI (*Graphical User Interface*) i demés diagrames UML de la primera iteració.
- **Familiarització i adaptació:** Aquesta segona fase tractarà de familiaritzar-se amb els conceptes de PD, Swarm intelligence i computational ecologies. Adaptar-se a l'entorn de Visual Studio i el seu llenguatge Visual Basic amb les llibreries .NET.
- **Implementació:** El gruix del projecte estarà en aquest punt. S'implementarà l'aplicació basant-se en l'estudi fet en el primer apartat.
- **Testatge:** Aquest apartat podria ben ser un subapartat del punt anterior. Després de la implementació de cada mòdul, s'haurà de fer un conjunt de proves del mòdul en qüestió per veure si continuem amb la implementació del següent mòdul (o a penjar l'executable a la web, en cas de ser l'últim mòdul) o arreglem els errors d'aquest per no arrossegar-los durant el transcurs de l'aplicatiu.
- **Documentació final:** Finalment, s'haurà de plasmar la feina feta en aquesta documentació i repassar el contingut que s'hi ha afegit durant el transcurs de la implementació.

### 1.6.1 Diagrama de Gantt

A la següent pàgina veurem el diagrama de Gantt.

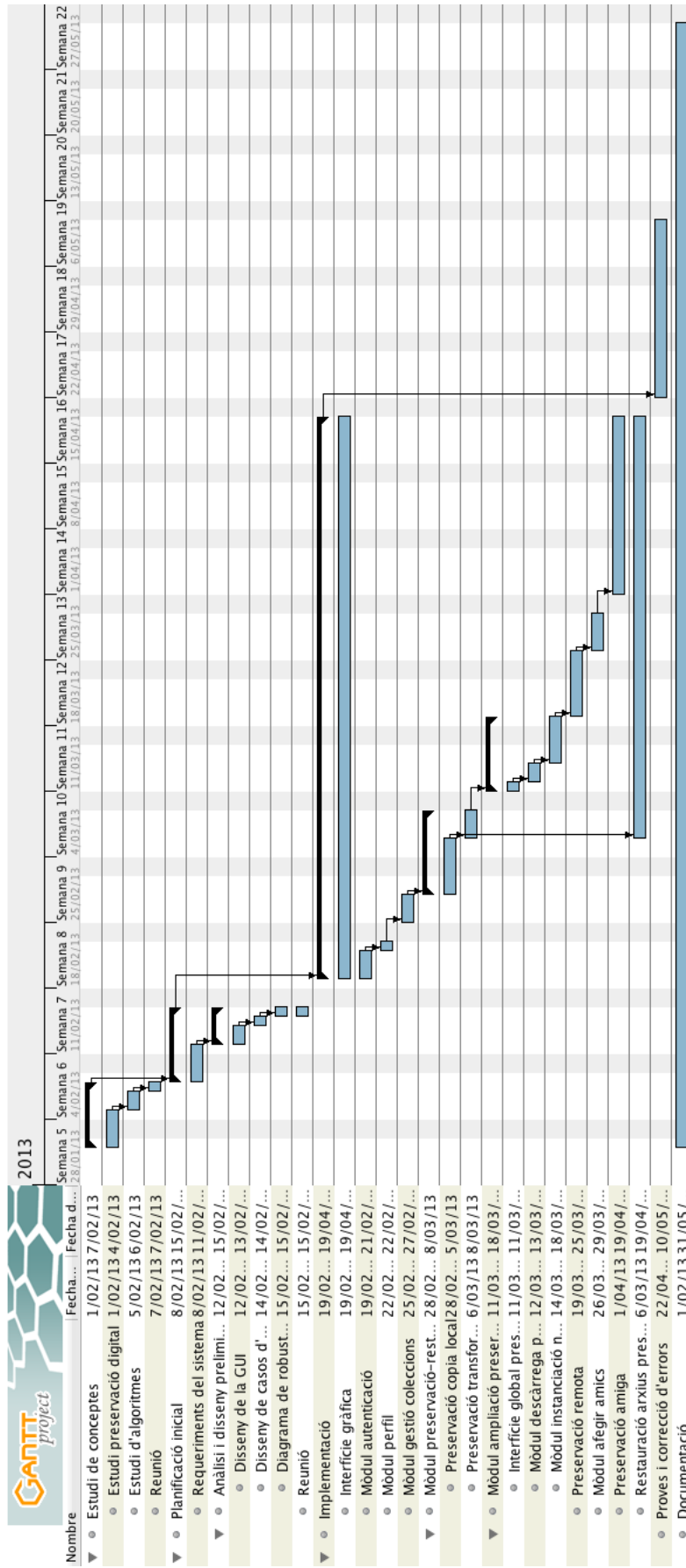


Figura 1.6: Diagrama de seqüència d'iniciar sessió.



## 1.7 Estructura de la documentació

- **Introducció:** En aquest primer apartat serveix per situar-te en l'àmbit on es du a terme, sables quins són els objectius, l'abast del projecte, la seva organització i estructura.
- **Conceptes previs:** Aquest projecte te a veure amb una part de la intel·ligència artificial i amb nous conceptes de Preservació digital així com algunes tècniques. Per tan, abans de continuar amb el transcurs d'aquest, s'han d'aclarir alguns conceptes importants per a poder entendre el projecte al complet.
- **Requeriments:** El projecte, abans de començar, ha de tenir uns requeriments, establerts pel client, del que es preten fer, quins mòduls tindrà i com han de ser. A més s'ha de decidir amb quin software es farà i quin hardware es necessitarà.
- **Anàlisi:** Un cop entesos els requeriments, objectius i abast, s'ha de començar amb l'anàlisi i l'enginyeria del software, en aquest capítol s'haurà de fer l'especificació i el prototipus de l'aplicació final, seguint la metodologia Iconix.
- **Disseny:** Un cop feta la part d'especificació, es segueix la part de disseny de l'enginyeria del software, quins patrons de programació es poden fer servir per a poder tenir la feina millor modulada i més ben estructurada, a més de facilitar certs aspectes de la programació.
- **Implementació i Testatges:** Quan ja s'han enllestit requeriments i plantejat l'anàlisi i el disseny, es comença a posar a la pràctica els coneixements adquirits durant la recerca i a donar forma al projecte i, finalment, fer proves per veure que els requisits es compleixen.
- **Conclusions:** Avaluem si s'han complert els objectius i requeriments que s'havien establert a l'inici del projecte i què es podria millorar.
- **Manual d'usuari:** Es fa una explicació de la instal·lació i la forma de fer servir l'aplicatiu desenvolupat.
- **Annexos:** En aquest apartat es reflexa el material afegit relacionat amb el projecte.

# Capítol 2

## Conceptes previs

Abans de continuar amb la documentació, s'han d'aclarir uns quants conceptes per tal de facilitar la lectura d'aquesta memòria i entendre millor les idees i aspectes que es donen a conèixer i que apareixen sovint durant el transcurs del projecte.

### 2.1 Intel·ligència artificial

La Intel·ligència Artificial (IA), es pot considerar com una de les disciplines més noves, essent considerada alhora com una gran desconeguda i una de les que desperta més interès. Això és degut a que poca gent té clar què és la IA, però no obstant això és considerada per una gran majoria de científics com la disciplina on han pensat alguna vegada en treballar. Existeixen una gran varietat de definicions respecte què és la IA, algunes més acceptades que d'altres, es podria resumir comentant que la IA tracta de "desenvolupar sistemes que pensin i actuïn racionalment".

En els últims anys la IA ha anat evolucionant, potser amb major celeritat que altres disciplines, motivat probablement per la seva pròpia immaduresa. Fet que ha dut a que la IA actualment abasti una gran quantitat d'àrees, des d'algunes molt generals com raonament, cerca, etc. a altres més específiques com els sistemes experts, sistemes de diagnòstic, etc. Podríem indicar, sense cap dubte, que la IA pot ser aplicada avui en dia a infinitat de disciplines científiques i és que la IA és susceptible a aparèixer allà on es requereixi d'intel·lecte humà. En aquest aspecte cada nova incursió de la IA en un camp diferent suposa emprar una nova metodologia d'aplicació pròpia de l'investigador que ho intenta, ja que a diferència d'altres disciplines, no existeixen certs estàndards o criteris consensuats que unifiquen el procés d'aproximació de la IA a altres àrees i, en conseqüència, a problemes reals. L'aparició d'una metodologia comuna a l'hora d'abordar l'aplicabilitat de la IA pot resultar un element positiu en la curta vida d'aquesta disciplina.

En l'àmbit de la IA va sorgir un nou paradigma conegut com "paradigma d'agents", el qual està prenent un gran auge entre els investigadors. Aquest nou paradigma aborda el desenvolupament d'entitats que poden actuar de forma autònoma i raonada. Si reprenem la definició donada anteriorment on es considerava a la IA com un medi pel desenvolupament de sistemes que pensen i actuen racionalment, podem pensar que la IA, en el seu conjunt, tracta realment de construir precisament dites entitats autònomes i intel·ligents.

D'acord amb aquesta visió, es pot considerar a la IA com una disciplina orientada a la construcció d'agents intel·ligents on s'integren les diferents àrees que aquesta comprèn. Des de fa relativament poc temps aquesta idea està sent adoptada per investigadors en la matèria; el propi Stuart Russell [5] per donar un enfocament de la IA orientat totalment al desenvolupament d'agents. És important ressaltar que en el moment en que es disposi d'una metodologia clara per la construcció d'aquests agents intel·ligents, l'aplicabilitat de les tècniques de IA a qualsevol tipus de problema podrà abordar-se d'una manera més clara i unificada.

De moment, es pot assegurar que, avui en dia, existeix una idea generalitzada dins de la IA de que es trobem davant el naixement d'una nova tecnologia, la d'**agents intel·ligents**, que permet abordar d'una manera més apropiada la construcció de sistemes intel·ligents més complexos aplicats a diversos camps. Vegem en els següents punts del document una descripció més detallada del que és u el que promet aquesta nova tecnologia cap a la que està orientada actualment la IA.

### 2.1.1 El concepte d' *Agent intel·ligent*

La creixent investigació en el camp de la IA, i més concretament en la Intel·ligència Artificial Distribuïda (IAD) ha dut a afrontar problemes complexos a través dels avantatges que suposa la utilització d'agents.

Un dels primers apropaments al terme agent va ser proposat l'any 1982 per Newell [6], que considerava que en el disseny de sistemes existien dos nivells d'abstracció: el nivell de símbols i el nivell de coneixement. En el nivell de símbols, es contemplen aquells programes convencionals que només manegen símbols (expressions o variables). Per altra banda, en el nivell de coneixement, un agent, equivalent a un programa en aquest nivell, maneja coneixement. D'aquesta forma, un sistema en el nivell de coneixement de Newell es pot considerar un agent. Així, aquest agent està format per una estructura, unes accions que aquest pot realitzar i unes metes a les que desitja arribar, i a més es comporta d'acord amb el principi de racionalitat que determina quines accions ha de realitzar per arribar a les seves metes. Actualment, existeixen moltes definicions del terme agent [7] [8] [9] [10] encara que cap és acceptada per unanimitat. Tot i això, la definició de Wooldridge i Jennings [11], és la que pren més força: "agent és un sistema informàtic que està situat en un entorn

i que té autonomia per tal d'aconseguir els seus objectius de disseny" (veure figura 2.1).

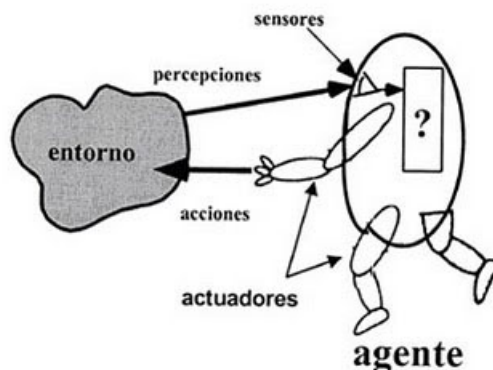


Figura 2.1: Visió esquemàtica d'un agent intel·ligent

Aquesta definició ens dona dues característiques bàsiques. En primer lloc l'autonomia indica que els agents tenen la capacitat de prendre decisions de manera independent, cosa que contribueix a la descentralització del sistema que els empra; en segon lloc, el fet d'estar situats fa referència al tipus d'entorns on els agents tendeixen a ser usats, que solen ser dinàmics, impredecibles i poc fiables. Com a dinàmics s'entén que canvien ràpidament; com a impredecibles, que no es poden predir els següents estats de l'entorn; i com a poc fiables, que les accions dutes a terme pels agents poden fallar per motius que escapen del seu control.

A causa de les característiques dels entorns, necessiten noves propietats. Els entorns dinàmics, creen la necessitat que esdevinguin reactius per respondre a temps als canvis que s'hi donin.

Els agents tenen la capacitat de tenir un comportament orientat a complir objectius i prendre la iniciativa, que s'anomena proactivitat. Una propietat d'aquests objectius és la seva persistència, cosa que els dóna robustesa, ja que seguiran intentant assolir l'objectiu malgrat els intents fallits. La proactivitat és una propietat clau en la distinció dels objectes i dels agents, ja que els objectes no poden tenir múltiples objectius, ni explícits ni persistents. Un punt clau en el disseny és el balanç entre la reactivitat i la proactivitat.

Com que l'entorn pot ser poc fiable, els agents han de ser robustos per recuperar-se de les fallades. Una manera natural d'aconseguir-ho és mitjançant la flexibilitat que permetrà a l'agent aconseguir un objectiu mitjançant diferents camins. A més, els agents interactuen amb altres agents (i potser amb persones), a través d'algun llenguatge de comunicació, de manera que són socials. Aquesta interacció normalment és a alt nivell utilitzant *performatives*[11]. Per aquests motius, una altra definició

d'agent és: "Porció de software situada, autònoma, reactiva, proactiva, flexible, robusta i social".

Cal constatar, que l'agència forta també inclou altres propietats com ara les creences, els desitjos i les intencions (BDI).

#### **2.1.1.1 Els agents com a metàfora de disseny**

Els agents proporcionen als dissenyadors i desenvolupadors de software una forma d'estructurar una aplicació al voltant de components autònoms i comunicatius, i una forma de dirigir la construcció d'eines de software i infraestructura de suport a aquesta estructura de disseny. En aquest sentit, els agents ofereixen una nova forma de dissenyar, més apropiada per al desenvolupament de sistemes computacionals complexos, especialment en entorns oberts i dinàmics.

Hi ha diverses metodologies d'anàlisi i disseny pròpies dels agents, entre aquestes MESSAGE-INGENIAS, ZEUS, TROPOS i GAIA. El problema que tenen aquestes metodologies és que presenten nombroses carències. També s'han proposat arquitectures agent per al disseny de components individuals de software, encara que cap, ara com ara, s'està imposant. La més famosa és RETSINA (Reusable Environment for Task-Structured Intelligent Networked Agents)[13]. L'entorn de suport distribuït, que tot i les seves nombroses carències s'està imposant per a la programació d'agents en Java, és JADE. Finalment es necessita infraestructura de suport per les activitats dels agents, que abasta altres tecnologies com per exemple els Web services.

#### **2.1.1.2 Els agents com a simulació**

Els agents ofereixen models potents per representar entorns reals complexos i dinàmics (canviants). Per exemple, la simulació de models econòmics, societats i entorns biològics són les àrees típiques d'aplicació.

L'ús d'agents per simular dominis del món real pot donar respostes als complexos problemes físics o socials que d'altra manera seria impossible simular degut a la complexitat involucrada. La simulació basada en agents s'estén a les estructures socials i institucions per desenvolupar explicacions plausibles dels fenòmens observats; a l'ajuda en el disseny d'estructures organitzatives, i donar llum sobre com funcionaran les polítiques o decisions de gestió, sistemes físics, incloent els edificis intel·ligents, tràfic de sistemes i poblacions biològiques; i els programes informàtics de tot tipus, que actualment inclouen comerç electrònic i els sistemes de gestió de la informació.

A més, els agents poden ser utilitzats per simular el comportament dels sistemes informàtics complexos. Aquests models de simulació poden ajudar als dissenyadors

i desenvolupadors de complexos sistemes de còmput i proporcionar orientació als enginyers de software responsables del control operatiu d'aquests sistemes. Els models de simulació basats en agents, per tant, donen solucions per gestionar millor els entorns on-line d'assignació de recursos a gran escala.

### 2.1.2 Sistema multiagent

Donat el caràcter distribuït que tenen els entorns en els que s'apliquen agents, resulta adequat considerar grups d'agents que interaccionen entre si per aconseguir objectius comuns. Segons la ObjectManagementGroup<sup>1</sup>, un sistema multiagent (SMA) pot definir-se com: “una plataforma que pot crear, interpretar, executar, transferir i finalitzar agents”.

Els SMA, com diuen J.M. Corchado i J.M. Molina [12], són sistemes computacionals en els que diversos agents autònoms interaccionen entre si; ja sigui per col·laborar en la solució d'un conjunt de problemes o en la consecució d'una sèrie d'objectius individuals o col·lectius. Aquests agents poden ser homogenis o heterogenis, poden tenir metes comuns o no i poseeixen, en la majoria dels casos, algun grau de comunicació entre ells. Cadascun d'aquests agents podrien tenir comunicació directa amb éssers humans a través d'interfícies o agents d'usuari.

Com explica K.P. Sycara[13], es destaquen les següents característiques del SMA:

- Cada agent té informació incompleta o un punt de vista limitat per resoldre el problema.
- No existeix un control global per tot el sistema.
- Les dades estan descentralitzades.
- La computació és asíncrona.

Per altra banda, tal i com s'indica a [13], els SMA es caracteritzen per la seva modularitat. Si un domini és particularment complex, ampli o impredecible, aquest pot ser simplificat mitjançant el desenvolupament d'un número de components (agents) funcionals, específics i modulars. Aquesta descomposició permet a cada agent emprar el paradigma més apropiat per resoldre el seu corresponent problema. No obstant això, degut a aquesta divisió, tot agent ha de ser capaç de coordinar-se amb altres per resoldre problemes d'interdependència.

La tecnologia de SMA està realitzant importants aportacions en la resolució de problemes en diversos dominis on aproximacions tradicionals no proporcionen solucions suficientment satisfactòries; comerç electrònic, telemedicina i un llarg etcètera.

---

<sup>1</sup>Consorti dedicat a mantenir i establir diversos estàndards de tecnologies orientades a objectes. Web: <http://www.omg.org>

## 2.2 Preservació Digital (PD)

### 2.2.1 Què és la PD?

Actualment, el gruix de la informació neix en format digital, i molta és disseminada exclusivament en aquest format. Tot i això, pocs objectes digitals tenen mitjans per assegurar la intel·ligibilitat de manera perpètua [14]. La informació en suports de paper i altres formats analògics no es pot copiar sense error i sempre conté variacions accidentals, cosa que les còpies en formats digitals eviten. Per contrapartida els suports analògics tenen una vida més llarga i la seva degradació és més lenta que els digitals.

Hi ha un ampli ventall d'institucions i d'usuaris particulars de diverses àrees amb interessos de preservar els seus objectes digitals durant les properes dècades, alguns exemples són les companyies farmacèutiques i automobilístiques, a les quals els interessa mantenir les seves dades, estudis i simulacions; i usuaris particulars que volen preservar les seves col·leccions de fotografies i de fitxers de vídeo i d'àudio [15]. A més, cal constatar que algunes institucions tenen obligacions legals de preservar alguns dels seus objectes digitals.

La preservació digital, que va aparèixer al 1995 [16], és la necessitat de mantenir l'accessibilitat amb un seguit de propietats dels objectes digitals, concretament l'autenticitat i la integritat al llarg del temps. Aquesta necessitat ha nascut com a conseqüència de la tècnica, l'evolució constant del maquinari, del programari i dels formats dels fitxers i la sindicació de continguts.

Segons Gladney [14], una solució completa de preservació digital, ha de complir els següents punts:

- Assegurar que una còpia de cada registre preservat sobreviu tant de temps com es desitgi.
- Assegurar que els consumidors autoritzats poden trobar i utilitzar qualsevol registre preservat, tal i com pretenien els seus productors i excloent els errors introduïts per terceres entitats.
- Assegurar que cada consumidor pugui decidir si la informació rebuda és suficientment fidedigne.
- Amagar la complexitat tecnològica als usuaris, tant productors com consumidors.

D'aquest punts ens sortiran els requeriments funcionals 3,4 i 5, així com el 6 dels no funcionals.

## 2.2.2 Estratègies de PD

Les estratègies de preservació actuals es poden classificar en dues aproximacions principals:

- **La preservació de l'entorn tecnològic.** Mentre preservem l'entorn en que els objectes digitals són accessibles, aquests ho seguiran essent. Per aconseguir aquesta aproximació existeixen dues tècniques:
  - *La preservació de la tecnologia.* Consisteix en preservar rèpliques de tot el maquinari i les plataformes de programari pel seu ús futur.
  - *L'emulació de la tecnologia.* Permet que els nous sistemes informàtics emulin sota demanda el hardware i el programari obsolet.
  
- **Superar l'obsolescència dels formats de fitxers.**
  - *La migració.* Consisteix en modificar el format de l'objecte de manera que sigui independent del software i del hardware necessari per crear-lo [16].
  - *L'encapsulació.* Consisteix en preservar l'objecte digital i tot allò necessari per garantir-ne l'accés [17].

A continuació es descriu cadascuna de les estratègies de preservació digital amb més detall.

### 2.2.2.1 Preservació de la tecnologia

Aquesta estratègia pretén disposar de museus que continguin equipament obsolet que es mantindran amb l' objectiu de replicar qualsevol configuració antiga de hardware i software. Per mantenir el context tecnològic d' accessibilitat dels documents digitals és necessari *çongelar* l'estat de la tècnica en un moment determinat. Això implica la preservació del programari original, a més del sistema operatiu i el maquinari. Els que promulguen aquesta estratègia, denoten la importància de mantenir l' entorn per conservar l' aspecte i les sensacions que transmet l' objecte digital original. Alguns inconvenients d' aquesta aproximació són la necessitat d' espai físic, el manteniment i el cost econòmic, la qual cosa la converteixen en una estratègia impracticable a llarg termini.



### 2.2.2.2 Emulació de la tecnologia

Aquesta aproximació té molts punts en comú amb l'anterior, però no inclou el manteniment del maquinari i del sistema operatiu de l'objecte original. La idea és que els emuladors, que són executats en màquines noves, estiguin programats per imitar el comportament del sistema original. L'objectiu d'aquesta estratègia és preservar l'aparença i la manera de funcionar de l'objecte, com també les seves funcionalitats [18].

Hendley, veu l'emulació com una estratègia de curt a mig termini, o si més no com una estratègia per especialistes on la necessitat de mantenir l'aspecte físic dels objectes és de molta importància pels usuaris. Whaugh [17], també apunta que l'aplicació pot contenir virus que causin la pèrdua d'informació i a més considera que serà útil si l'objectiu és preservar el programari com un artefacte, i si en el seu futur les organitzacions no tenen prou coneixement com per entendre el format de la informació digital. Granger[19] expressa que l'emulació és una solució parcial a la preservació digital, mentre que alguns autors com Russell[20], Woodyard[21], Gilheany[22] i Rothemberg [18] [23] veuen aquesta estratègia com la de més potencial per a la preservació a llarg termini.

Cal esmentar que com l'emulació és una estratègia que necessita una especificació detallada del maquinari obsolet i del sistema operatiu, es necessita la generació d'estàndards. De moment, encara està obert el debat de si els emuladors han d'imitar l'entorn de manera completa, o només els aspectes més rellevants per accedir a les dades [24].

### 2.2.2.3 Migració de la informació

Aquesta tècnica s'ha imposat com la solució més acceptada per la creació i el manteniment de fitxers digitals i és la que emprarem en el nostre projecte. Parteix de la idea de que els documents han de ser accessibles a partir dels sistemes informàtics existents en cada moment, el que exigeix una migració periòdica a formats intel·ligibles pels sistemes actuals.

El *Consultative Committee for Space Data Systems* (CCSDS) va desenvolupar el *The Open Archiving Information System* (OAIS), que divideix la migració en quatre categories:

- **El refresc:** S'encarrega de mantenir una còpia de la cadena de bits de l'objecte
- **La replicació:** S'encarrega de que hi hagi una còpia de l'objecte disponible
- **El reemplaçament:** S'encarrega de que es sobreescrigui l'objecte disponible
- **La transformació:** S'encarrega de transformar la cadena de bits de l'objecte.

La migració de la informació es basa en poder accedir al contingut intel·lectual mitjançant la tecnologia actual. Aquesta estratègia podria ser aplicada més fàcilment si els objectes digitals fossin migrats a un conjunt petit de formats estàndard que fossin independents del hardware i del software, copiant els objectes a formats analògics i a aplicacions amb compatibilitat cap enrere [25]. La compatibilitat cap enrere del programari, pot resoldre el problema de la preservació a curt termini.

Hendley i Berman creuen que aquesta estratègia és la més prometedora. Russell creu que és la més pràctica, com a mínim pel curt i mig termini, i que a llarg termini el cost de la migració excedirà els costos de preservar la tecnologia o el de l'emulació. Tot i això, en els objectes amb múltiples components cal fer una migració d'aquests per separat, cosa que pot resultar realment complicat.

D'aquest apartat ens surten els requeriments funcionals 3 i 4 de la nostra aplicació.

#### 2.2.2.4 Encapsulació

L'encapsulació espera superar els problemes d'obsolescència del format de fitxers encapsulant la informació de la interpretació de l'objecte. Aquesta estratègia inclou crear l'aplicació que es va utilitzar per crear i/o accedir a l'objecte en les plataformes futures. Part del procés inclou migrar l'objecte a un format més documentat. Rothemberg ha promogut molt l'encapsulació, i Shepard i Day recolzen l'estratègia. D'altra banda Bearman defensa que no està clar com es pot implementar.

## 2.3 Swarm intelligence

Aquesta tècnica està ubicada dins dels *algorismes de computació evolutiva*[26].

El terme intel·ligència d'eixam va sorgir a finals de 1990 a la intel·ligència artificial, la robòtica, la vida artificial i a comunitats de resolució de problemes distribuïts, inspirant-se en l'observació de colònies d'insectes socials. Una definició comunament acceptada és que la característica d'un sistema pel qual els comportaments col·lectius (poc sofisticats) d'agents que interactuen a nivell local amb l'entorn condueixen a l'emergència d'un comportament global complex [27][28].

Aquests sistemes estan conformats per una població d'agents computacionals simples capaços de percebre i modificar el seu ambient de manera local. Aquesta capacitat fa possible la comunicació entre els individus, que detecten els canvis en l'ambient generat pel comportament dels seus semblants. Encara que normalment no hi ha una estructura centralitzada de control que dictamina com els agents han de comportar-se les interaccions locals entre els agents usualment porten a la emergència d'un comportament global. Una altra característica addicional és la inexistència d'un model explícit de l'ambient.

Per tant, hi ha bones raons per trobar la intel·ligència d'eixam atractiva: els sistemes de programari actuals són tan complexos i difícils de resoldre que ja no poden ser controlats, i la intel·ligència d'eixam ofereix una alternativa de disseny en que l'autonomia, l'emergència i el funcionament distribuït reemplaça el control, la pre-programació i la centralització.

### 2.3.1 La colònia de les formigues

El paradigma principal de la intel·ligència d'eixam és la colònia de formigues. En ella, el comportament de formigues individuals és controlada per un petit conjunt de regles molt simples, però les seves interaccions amb l'entorn els permet resoldre problemes complexos, com ara trobar el camí més curt d'un punt a un altre. Les colònies de formigues són sistemes intel·ligents, amb gran capacitat de resolució de problemes, formades a partir d'una quantitat de subsistemes relativament independents i simples que no mostren la intel·ligència individual.

La optimització per colònia de formigues (ACO) és una família d'algorismes derivats del treball realitzat per Dorigo [29], basada en el comportament social de les formigues, les quals utilitzen una forma de comunicació basada en substàncies químiques denominades feromones. Aquestes substàncies, dipositades per la formiga a l'avançar per un camí, exerceixen una acció sobre la decisió de les formigues precedents, les quals escullen el camí que posseeixi una major concentració de substància, permetent que trobin la ubicació de les fonts d'aliment així com el seu niu. S'ha demostrat que els rastres de feromona permeten lentament l'optimització distribuïda en la qual cada agent senzill realitza una petita distribució en la cerca de la millor solució.

En els algorismes ACO cada agent construeix una solució o part d'aquesta, començant per un estat inicial i desplaçant-se a través d'una seqüència finita d'estats veïns, fent us de dos fonts d'informació: la visibilitat i els rastres de feromona. La probabilitat de que la  $k$ -èsima formiga es desplaci del node  $i$  al node  $j$  està donada per l'equació 2.1, sent els nodes  $J_i^k$  els estats vàlids. Al mateix temps, els rastres són modificats per canviar la representació del problema, que utilitzaran les altres formigues per prendre les seves decisions per mitjà de l'equació 2.2, on és el coeficient "d'evaporació" de la feromona emprat per evitar la ràpida convergència de les formigues cap a una regió de l'espai de cerca. Llavors la feromona es converteix en una memòria local compartida de llarg termini que influencia les decisions subsequents de les formigues.

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (2.1)$$

$$\tau(t + 1) = \rho \cdot \tau(t) + (1 - \rho) \cdot \Delta\tau(t, t + 1) \quad (2.2)$$

Les formigues poden actuar concurrent o independentment, mostrant un comportament cooperatiu que emprà la stigmergia, una forma de comunicació indirecta per mitjà de la modificació de l'ambient. Encara que cada una de les formigues és capaç de trobar una solució, probablement inadequada, les millors solucions són trobades com el resultat de la cooperació global entre tots els agents de la colònia, com una característica inesperada de la interacció cooperativa dels agents.

Altres característiques usualment incloses dins d'un sistema ACO són el balanç entre les característiques aleatòries i la severitat en la presa de decisions (exploració del coneixement); i estratègies de coneixement global denominades *daemon actions*. Les característiques d'un algorisme ACO poden necessitar alguna classe de sincronització, usualment obtinguda per una programació seqüencial, on el coneixement global és fàcilment accessible en qualsevol instant.

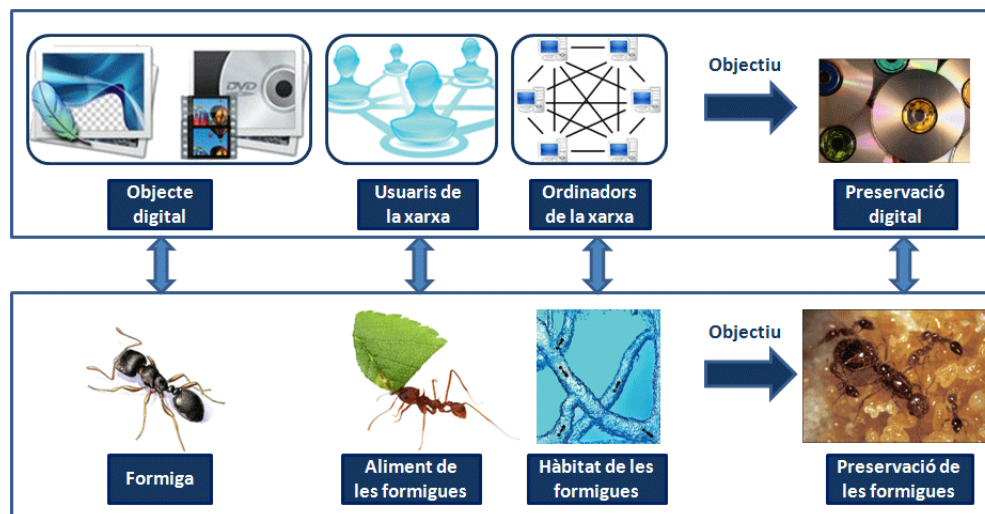


Figura 2.2: Analogia entre la preservació digital i la preservació de les formigues.

## 2.4 Computational ecologies

Es descriu com una forma de computació distribuïda<sup>2</sup> on els agents tenen coneixements incomplets i informació imperfecte de l'estat del sistema. Quan els agents poden escollir entre diversos recursos la dinàmica del sistema pot ser oscil·latòria i fins i tot caòtica. Llavors es descriu un mecanisme per a una estabilitat global a través de controls locals. [30]

Com s'explica a [30], impulsada pels avanços en disseny de software i l'increment de la connectivitat de les xarxes, els sistemes de computació distribuïda es comencen a estendre a través d'oficines, laboratoris, països i continents. En aquests processos de sistemes computacionals que consisteixen en activar l'execució de programes, poden generar d'altres en altres màquines ja que fan ús d'impresores, servidors i altres altres màquines de la xarxa quan sorgeix la necessitat. En les aplicacions més complexes, poden col·laborar diversos processos per a solucionar problemes mentre competeixen pels recursos disponibles al mateix temps que poden interactuar amb el món físic. Això contrasta amb els mètodes tradicionals de planificació centralitzada per a l'assignació de recursos i mètodes de programació basants en processament en sèrie dels més familiars ordinadors/equips independents.

L'efectivitat en l'ús de la computació distribuïda és una tasca complicada, ja que els processos han d'obtenir recursos en un entorn que canvia dinàmicament i han d'èsser dissenyats per col·laborar entre ells tot i que aquest entorn té una varietat de canvis impredecibles i asíncrons. Per exemple, la manca de perspectives globals per determinar l'assignació de recursos requereix un enfocament molt diferent a la programació a nivell de sistema i a la creació de llenguatges adequats. Així doncs, és difícil implementar mètodes adequats que mitjançant els seus processos es pugui computar en màquines amb diverses característiques.

A mesura que aquests sistemes distribuïts creixen, esdevenen una comunitat de processos concurrents, o un **computational ecosystem** [32] (ecosistema computacional), que en les seves interaccions, estratègies, i mancança de coneixement perfecte, són anàlegs dels ecosistemes biològics i economies humanes. Atès que tots aquests sistemes consisteixen en un gran nombre d'actors independents competin pels recursos, aquesta analogia pot suggerir nous camins per concebre el comportament d'aquests sistemes computacionals emergents. En particular, aquests sistemes existents tenen mètodes per fer front satisfactòriament amb operacions coordinades asíncronament de cares el coneixement imperfecte. Aquests mètodes permeten al

---

<sup>2</sup>Un sistema distribuït es defineix com una col·lecció de computadores separades físicament i connectades entre si per una xarxa de comunicacions distribuïda; cada màquina poseeix els seus components de *hardware* i *software* que l'usuari percep com un sol sistema (no necessita saber quines coses estan en quines màquines). L'usuari accedeix als recursos remots (RPC) de la mateixa forma en que accedeix a recursos locals, o un grup de computadores que usen un *software* per aconseguir un objectiu en comú.[31]

sistema en el seu complet a adaptar-se als canvis en l'entorn o les pertorbacions dels membres individuals, en contrapartida a la fragilitat de la majoria de programes dels ordinadors actuals que acostumen a fallar si hi ha la més mínima variació de les entrades o errors en el mateix programa. És d'interés explotar aquesta analogia per a millorar la fiabilitat i la usabilitat de la computació distribuïda.

El rendiment dels processos computacionals poden ser compensats de diferents formes. Un mètode particularment atractiu i que es tractarà a continuació és imitar el mecanisme que es troba en l'**evolució biològica**, on l'aptitud determina el nombre de supervivents d'una espècie determinada en un entorn en canvi. Aquest mecanisme es usat en computació sota el nom de algorismes genètics[33].

## 2.4.1 Algorismes evolutius i algorismes genètics

Els algorismes evolutius corresponen a un grup de tècniques estocàstiques que utilitzen els conceptes d'**evolució biològica**. Els EA (*Evolutionary Algorithm*) actuen sobre una població de solucions potencials aplicant els principis de diversitat d'individus i supervivència del més fort per produir millors aproximacions a una solució. En cada generació és creat un nou grup d'aproximacions pel procés de selecció d'individus d'acord amb el seu nivell d'"acompliment" en el domini del problema, i es creuen entre si emprant operadors que imiten els conceptes genètics. Aquest procés porta a l'evolució de poblacions d'individus que estan millor adaptats a l'ambient.

### 2.4.1.1 Algorismes genètics

Els Algorismes genètics són mètodes adaptatius, generalment emprats en problemes de cerca i optimització de paràmetres, basats en la reproducció sexual i en el principi de supervivència del més apte.

Més formalment, i seguint la definició de Goldberg, qui va establir els principis bàsics, "Els Algorismes Genètics són algorismes de cerca basats en la mecànica de selecció natural i de la genètica natural. Combinen la supervivència del més apte entre estructures de seqüències amb un intercanvi d'informació estructurat, encara que aleatoritzat, per construir així un algorisme de cerca que tingui alguna cosa de les genialitats de les cerques humanes" [34].

Per obtenir la solució a un problema es parteix d'un conjunt inicial d'individus, anomenat població, generat de manera aleatòria. Cadascun d'aquests individus representa una possible solució del problema. Aquests individus evolucionaran prenent com a base els esquemes proposats per Darwin[35] sobre la selecció natural, i s'adaptaran en major mesura després del pas de cada generació a la solució requerida. L'evolució d'aquestes solucions cap a valors òptims del problema depèn en gran mesura d'una adequada codificació d'aquestes.

El poder dels Algoritmes Genètics prové del fet que tracta una tècnica robusta, i pot tractar amb èxit una gran varietat de problemes provinents de diferents àrees, incloent aquells en els que altres mètodes troben dificultats. Tot i que no es garanteix que els Algoritmes Genètics trobin una solució òptima del problema, existeix una evidència empírica de que es troben solucions d'un nivell acceptable, en un temps competitiu amb la resta d'algorismes d'optimització combinatòria.

## Bases biològiques

En la natura els individus d'una població competeixen entre si cercant recursos com menjar, aigua i refugi. Inclòs els membres d'una mateixa espècie competeixen sovint en la cerca d'un company. Aquells individus que tenen més èxit en la supervivència i en atraure companys, tenen major probabilitat de generar un gran nombre de descendents; pel contrari, individus menys dotats, produiran menor número de descendents (tornem a l'analogia de la supervivència dels forts). Això significa que els gens dels individus millor adaptats es propaguen en successives generacions cap a un número d'individus creixent. La combinació de bones característiques provinents de diferents ancestres, pot a vegades produir descendents "superindividus", l'adaptació dels quals és molt major que la de qualsevol dels seus ancestres. D'aquesta manera, les espècies evolucionen assolint unes característiques cada cop millor adaptades a l'entorn en el que viuen.

Els Algoritmes Genètics usen una analogia directa amb el comportament natural. Treballen amb una població d'individus, cadascun dels quals representa una solució factible a un problema donat. A cada individu se li assigna un valor o puntuació, relacionat amb la bondat d'aquesta solució. En la naturalesa això equivaldria al grau d'efectivitat d'un organisme per a competir per uns determinats recursos. Com més gran sigui l'adaptació d'un individu al problema, major serà la probabilitat que el mateix sigui seleccionat per a reproduir-se, creuant el seu material genètic amb altre individu seleccionat d'igual forma. Aquest creuament produirà nous individus - descendents dels anteriors - els quals comparteixen algunes de les característiques dels seus pares. Com menor sigui l'adaptació d'un individu, menor serà la probabilitat que aquest individu sigui seleccionat per a la reproducció, i per tant que el seu material genètic es propagui en successives generacions.

D'aquesta manera es produeix una nova població de possibles solucions, la qual reemplaça a l'anterior i verifica la interessant propietat que conté una major proporció de bones característiques en comparació amb la població anterior. Així, al llarg de les generacions les bones característiques es propaguen a través de la població. Afavorint el creuament dels individus millor adaptats, van sent explorades les àrees més prometedores de l'espai de cerca. Si l'Algorisme Genètic ha estat ben dissenyat, la població convergirà cap a una solució òptima del problema.

## Codificació de problemes

Qualsevol solució potencial a un problema pot ser presentada donant valors a una sèrie de paràmetres. El conjunt de tots els paràmetres (*gens* en la terminologia d'Algoritmes Genètics) es codifica en una cadena de valors denominada *cromosoma*. El conjunt dels paràmetres representats per un cromosoma particular rep el nom de *genotip*. El genotip conté la informació necessària per la construcció de l'organisme, és a dir, la solució real del problema, que s'anomena *fenotip*. Per exemple, en termes biològics, la informació genètica continguda en l'ADN d'un individu seria el genotip, mentre que l'expressió d'aquest ADN (el propi individu) seria el fenotip. Des dels primers treballs de John Holland la codificació es sol fer amb valors binaris [figura 2.3]. S'assigna un determinat número de bits a cada paràmetre i es realitza una discretització de la variable representada per cada gen. Tots els paràmetres no tenen que estar codificats amb el mateix número de bits. Cadascun dels bits que pertanyen a un gen sol rebre el nom d'*al·lel*. També existeixen representacions que

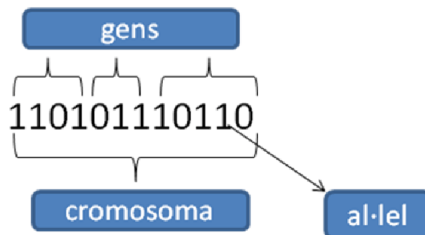


Figura 2.3: Individu genètic binari.

codifiquen directament cada paràmetre amb un valor enter, real, punt flotant, etc. Tot i que a aquestes representacions se'ls acusa de degradar el paral·lelisme implícit de les representacions binàries, permeten el desenvolupament d'operadors genètics més específics.



## 2.5 Community Cloud

Abans de començar a explicar el concepte de *Community Cloud*, hem d'introduir la idea de *Cloud Computing*.

### 2.5.1 Cloud Computing

S'anomena *Cloud Computing* a l'ús de les tecnologies basades en Internet per a la prestació de serveis[37], el terme *cloud*(núvol) és la forma en que, degut a la seva complexa forma, es representa l'internet.

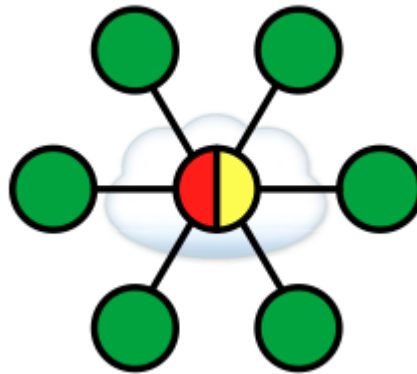


Figura 2.4: Cloud Computing

A la figura anterior podem veure la típica configuració dels clients visitant una aplicació servida per un núvol central que és situat en un o més centres de dades. Els nodes verds simbolitzen el consum de recursos i el node groc simbolitza la provisió d'aquests recursos, el node vermell representa la coordinació centralitzada de la provisió dels recursos.

### 2.5.2 Community Cloud Computing

*Community Cloud Computing* [38] apareix de la preocupació pel control dels proveïdors de *Cloud computing* i l'observació de que les preocupacions anàlogues impulsen a la investigació d'ecosistemes digitals.

Es vol arribar a combinar els principis d'ecosistemes digitals (*Computational Ecology*) amb els casos d'ús de *Cloud Computing*.

Així doncs, com veurem a la següent figura, substituïm els núvols de proveïdors

mitjançant la conformació dels recursos infrautilitzats de màquines dels usuaris per a formar un núvol comunitari amb nodes amb la possibilitat d'assumir tots els rols (consumidors, productors i, el més important, coordinadors).

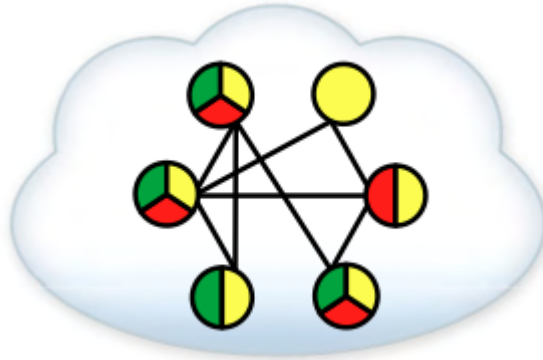


Figura 2.5: Community Cloud

Així doncs, podem concloure que el *Community Cloud* és una alternativa al *Cloud Computing*, creada a partir de la barreja d'escenaris d'usos amb els principis dels *Digital Ecosystems*.

Community Cloud Computing fa servir els recursos de diferents ordinadors personals connectats a la xarxa per a proporcionar els centres de dades, de tal manera que la comunitat ofereix la potència de càlcul per a la plataforma en el núvol que es vulgui emprar.

Aquest concepte, juntament amb el *swarm intelligence* i el *computational ecology* requereix d'una comunitat d'usuaris i, això dona lloc a l'últim punt dels requeriments funcionals (9).

## 2.6 Final del capítol

Un cop vistos aquests conceptes, es veu que necessitem d'una aplicació en versió alpha per a posar-los en pràctica, així doncs, juntament amb el contingut del capítol 1 (motivacions i objectius) ens surten uns requeriments per a l'aplicació a part dels requeriments que hem comentat durant la redacció d'aquest capítol que explicarem en el següent.

# Capítol 3

## Requeriments del sistema

En aquest apartat es veurà el què requerirà Pyramid i quin software s'empra per a la realització d'aquest. Hem de tenir en compte què necessita l'usuari: Una aplicació senzilla, fàcil d'usar i intuïtiva per a poder preservar el seu patrimoni digital. Per tan, li hem de donar una aplicació amb les següents característiques.

### 3.1 Requeriments funcionals

1. Autenticar l'usuari al sistema (Login).
2. Administrar col·leccions (àlbums).
3. Backups locals o remots de les col·leccions.
4. Transformacions locals o remotes de les col·leccions.
5. Restauració fidedigne de les preservacions.
6. Previsualització de l'arxiu.
7. Aplicar accions de preservació locals o remotes de forma intel·ligent.
8. Obtenir coneixement de la comunitat sobre els formats de moda.
9. Llista d'amics

### 3.2 Requeriments no funcionals

1. Interfície Multiidioma.

2. Actualitzacions automàtiques de l'aplicació
3. Possibilitat d'ampliació dinàmica de preservadors
4. Interfície senzilla, clara i d'estil 2.0. Fent ènfasis en la usabilitat.
5. Possibilitat d'executar-se en background amb un *tray icon* a la barra d'estat.
6. L'aplicació ha d'ésser lleugera, per tan, l'usuari no ha de notar cap relentiment del sistema quan aquesta estigui en funcionament
7. Base de dades compacta i portable
8. Informar de forma visual quan s'han realitzat operacions o s'han detectat nous arxius a les col·leccions.

### 3.3 Estudis i decisions

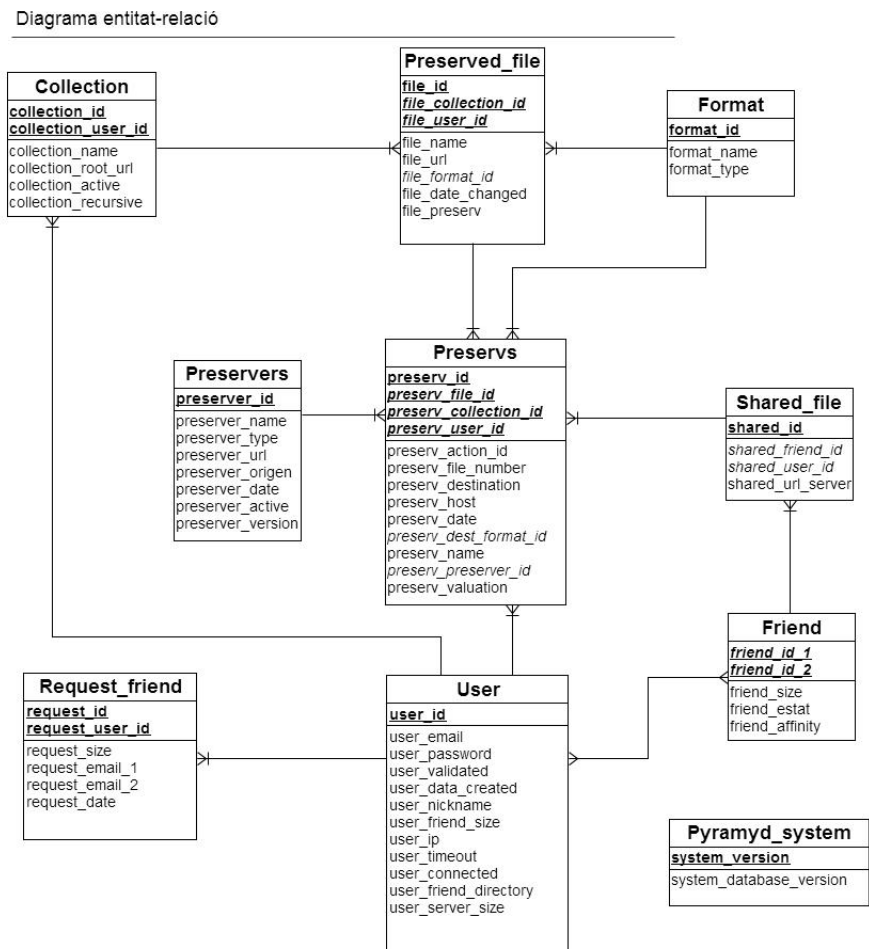
Per tal de donar senzillesa a la interfície, tenir una estructura de codi ben organitzada i mantenir una integritat del codi en tot moment, s'ha optat per seguir un patró en tres capes que manté separades les capes de negoci, dades i vista.

#### 3.3.1 Base de Dades (BD) i Servidor

Tots els mòduls han de compartir la mateixa base de dades, el mateix sistema d'autenticació, basat en usuari i contrasenya (que s'anirà propagant al llarg de la sessió d'usuari), les direccions d'on es situen els fitxers preservats i els originals i uniformar tant com sigui possible els criteris de disseny i la imatge visual.

La base de dades MySQL v5.0.90 està muntada sobre un servidor remot amb sistema operatiu *Solaris 10 10/09 s10x u8wos 08a X86* muntat amb un sistema de fitxers ZFS per a l'emmagatzematge de les dades (fotos), i per a la comunicació entre BD i servidor es fa a través d'un *WebService(ws) php*.

A continuació es mostra el diagrama entitat-relació de l'aplicació.



create and share your own diagrams at gliffy.com



Figura 3.1: Diagrama entitat-relació.

## 3.4 Software

En aquesta secció de la memòria es documenta tot el software emprat en la realització del projecte, així com les funcions que han proporcionat al desenvolupament.

### Microsoft Visual Studio 2010 Professional

Microsoft Visual Studio és un entorn de desenvolupament integrat (IDE, per les sigles en anglès) per a sistemes operatius Windows. Soporta diversos llenguatges de programació tals com Visual C++, Visual C#, Visual J# i Visual Basic . Net (aquest últim és el que s'ha escollit pel desenvolupament de Pyramid), tanmateix, al igual que entorns de desenvolupament web com ASP.NET.

Visual Studio permet als desenvolupadors crear aplicacions, llocs i aplicacions web, així com serveis web en qualsevol entorn que suporti la plataforma .NET. Així es poden crear aplicacions que s'intercomunicuin entre estacions de treball, pàgines web i dispositius mòbils.



Figura 3.2: Visual Studio 2010 Professional.

### MySQL

*MySQL* és un sistema de gestió de bases de dades relacionals, multi fil i multiusuari amb més de sis milions d'instal·lacions. Per una banda s'ofereix sota la llicència *GNU GPL* per a qualsevol ús compatible amb aquesta llicència, però per a aquelles empreses que vulguin incorporar-lo en productes privatis, és necessari adquirir una llicència específica que els permeti l'ús.

*MySQL* és multi plataforma i està desenvolupada en gran en *ANSI C*. A més té servidors de lliure descàrrega a la seva plana web oficial.



Figura 3.3: Logo de MySQL.

### Visual Basic .NET

*Visual Basic .NET* és un llenguatge de programació de Microsoft on estan incloses les llibreries .NET que ens serviran per a fer els preservadors. *.NET* és un component de software que pot ser afegit al sistema operatiu Windows. Proveeix un extens conjunt de solucions predefinides per a necessitats generals de la programació d'aplicacions, i administra l'execució dels programes escrits específicament amb la plataforma.



Figura 3.4: Visual Basic.



Figura 3.5: Logo de .NET.

## OmniGraffle

*OmniGraffle* és una eina de la suite OmniGroup.



(a) Logo de Omnigraffle

Figura 3.6: Software per a la creació de UML

Amb *OmniGraffle* s'han dissenyat tots els diagrames UML d'Iconix requerits pel desenvolupament del projecte.

## Gantt Project

*Gantt Project* és un software de lliure distribució per a realitzar diagrames de Gantt.



Figura 3.7: Logo de Gantt Project.



## Gliffy.com

*Gliffy.com* és un editor web de diagrames. Et permet crear y compartir diagrames de fluxe, diagrames de xarxa, planols, dissenys d'interfaç d'usuari, altres diagrames UML i altres dibuixos en línia.



Figura 3.8: Logo de Gliffy.

El *Gliffy* s'ha fet servir per a construir el diagrama entitat-relació.

## Visual SourceSafe 2005

*Visual SourceSafe 2005* és un controlador de versions per a Visual Studio. Aquest controlador et permet fer *rollbacks* (tornar a versions anteriors) i poder treballar amb la mateixa versió en diferents equips.



Figura 3.9: Logo de Source Safe 2005.

## XAML

*XAML* (*eXtensible Application Markup Language*, Lenguatge Extensible de Format per a Aplicacions) és el llenguatge de format per a la interfàç d'usuari per a la Base de Presentació de Windows (WPF) y Silverlight(wpf/e), el qual és un dels "pilars" de la interfàç de programació d'aplicacions .NET.

*XAML* és un llenguatge declaratiu basat en *XML*, optimitzat per a descriure gràficament interfícies d'usuari visuals riques des del punt de vista gràfic, per tan, s'ha fet servir per a implementar la interfície gràfica de l'aplicació.



Figura 3.10: Logo de XAML.

## FileZilla

*FileZilla* és un client FTP multiplataforma de codi obert i software lliure, llicenciat sota llicència pública general de GNU. Soporta els protocols FTP, SFTP y FTP sobre SSL/TLS (FTPS).



Figura 3.11: Logo de FileZilla.

Aquest software es fa servir per a guardar els nous preservadors, per a pujar els *webservices* de php i comprovar que els arxius es preserven en remot.

# Capítol 4

## Anàlisi

Tal i com s'ha vist al segon capítol (Conceptes previs), tenim una extensa teoria al darrera, però tot aquest fonament teòric precisa d'una plataforma mínima sobre la qual provar de forma empírica aquests fonaments.

Això implica una primera part d'interfície d'usuari i plataforma, i una segona part que deixa les portes obertes per aplicar recerca sobre els temes explicats i ampliar els resultats obtinguts d'aquest projecte.

### 4.1 Prototipus de GUI

En aquesta secció farem el primer pas de la metodologia que és crear un prototipus de la GUI (*Graphical User Interface*).

#### 4.1.1 Login

La pantalla de Login haurà de tenir els elements comuns (a la següent pàgina es pot veure la pantalla [4.1]):

- Identificació amb correu-e o ID i contrassenya
- Enllaç per a crear nou usuari
- Enllaç per a recordar contrassenya

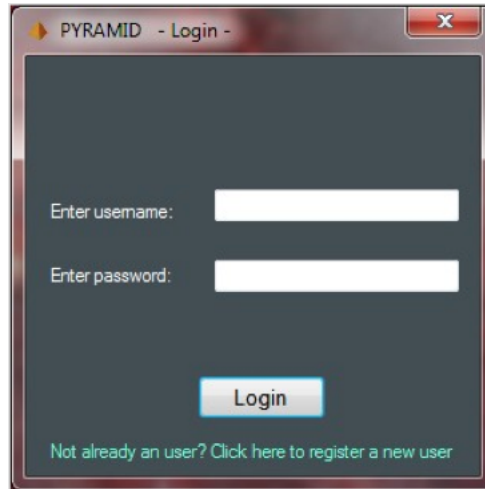


Figura 4.1: Pantalla de Login.

## 4.1.2 Pantalla principal

La pantalla principal haurà de permetre una fàcil navegació entre les diferents opcions així com informació visual sobre arxius afegits, àlbums creats, arxius preservats....

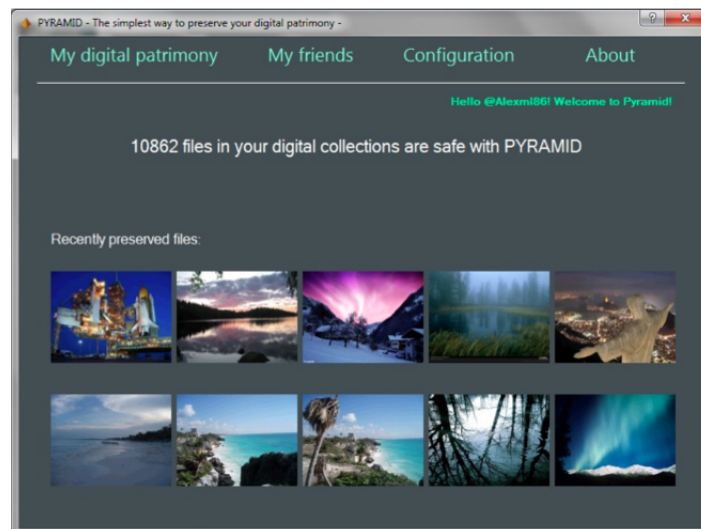


Figura 4.2: Pantalla principal.

### 4.1.3 Administració de patrimoni

Aquesta és la pantalla on veurem la informació de les nostres col·leccions i haurà de contenir:

- Navegar per les col·leccions o directoris
- Veure l'estat de preservació de cada arxiu
- Visualitzar informació sobre cada arxiu
- Previsualitzar els arxius
- Seleccionar arxius/col·leccions
- Eliminar/editar arxius
- Recuperar arxius preservats

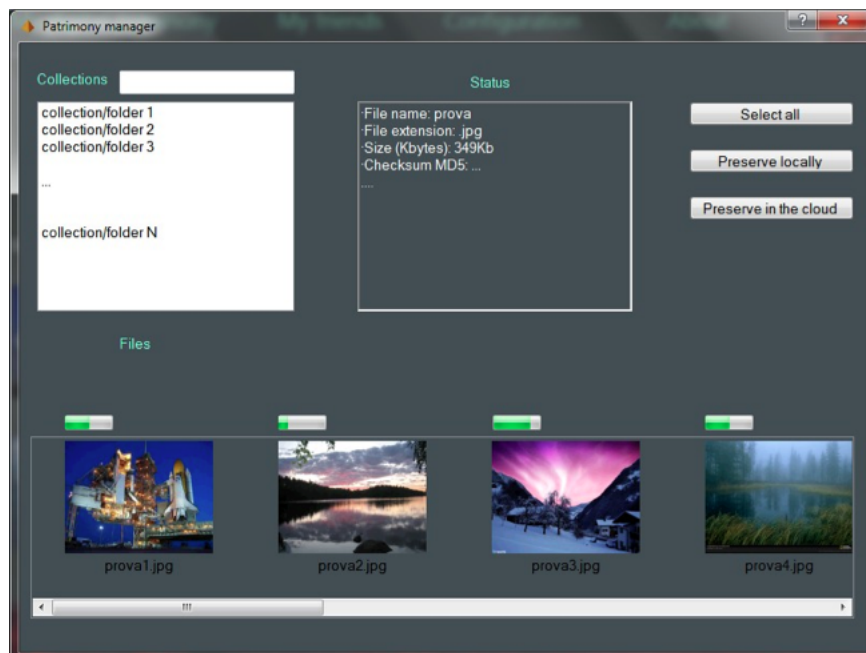


Figura 4.3: Administració de patrimoni.

#### 4.1.4 Afegir col·lecció

Aquesta vista haurà de permetre afegir col·leccions a partir de directoris, especificar nom de la descripció, guardar-la...

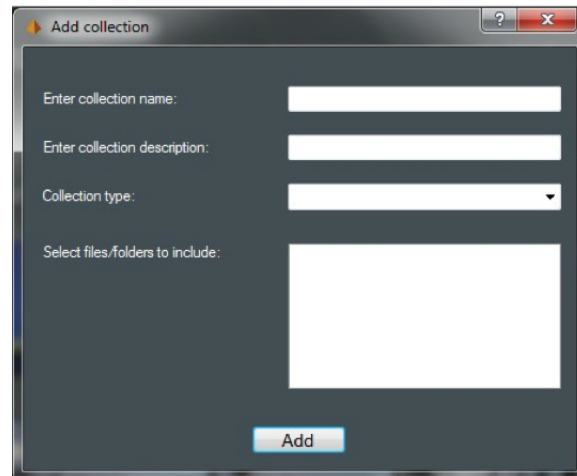


Figura 4.4: Afegir col·lecció.

#### 4.1.5 Pantalla de configuració

L'aplicació haurà de permetre configurar directoris a monitoritzar/escanejar i configurar, també, les preferències de l'usuari.

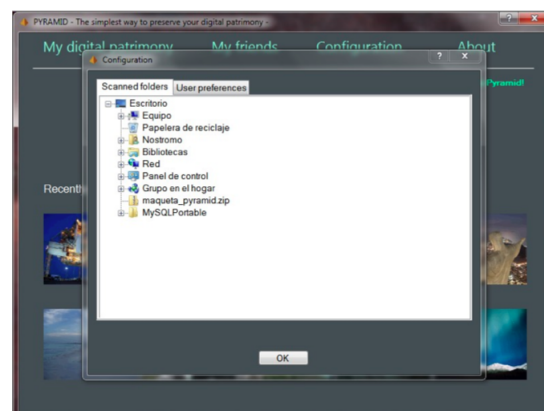


Figura 4.5: Pantalla de configuració.

### 4.1.6 Pantalla d'actualitzacions

L'aplicació haurà de permetre buscar, descarregar-se i instal·lar-se noves actualitzacions del programa.

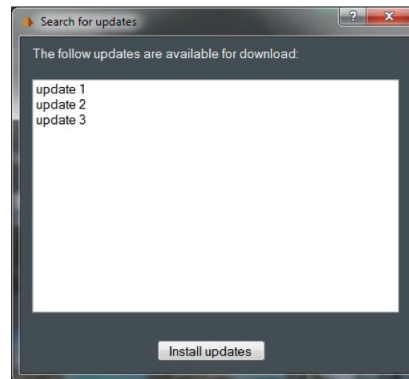


Figura 4.6: Pantalla de les actualitzacions.

### 4.1.7 Pantalla de *plugins*

Aquesta pantalla haurà de permetre buscar i instal·lar *plugins*, aquests estaran enllaçats a un patró factory per tal de carregar el que faci falta sense haver d'instanciar tots els *plugins* cada vegada.

Aquests *plugins* seran accions de preservació.

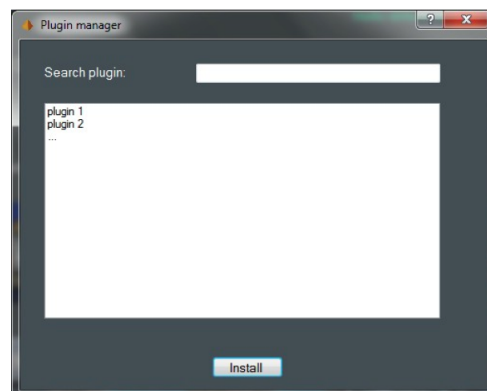


Figura 4.7: Pantalla dels plugins.

#### 4.1.8 *Tray icon*

Com s'ha dit als requeriments, l'aplicació s'haurà de poder executar en background, per tan, hi haurà una icona a la barra d'eines.

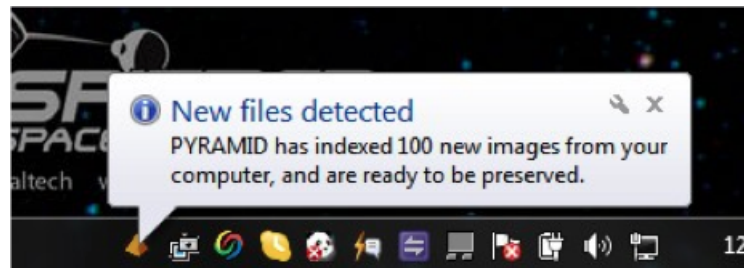


Figura 4.8: Tray icon.

#### 4.1.9 *Pantalla d'amics*

Aquesta pantalla haurà de permetre la gestió d'amics.

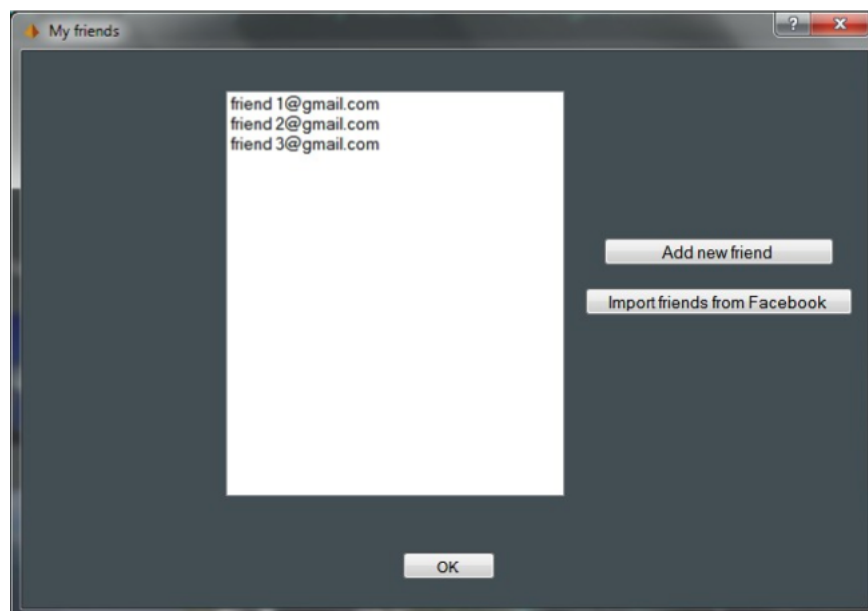


Figura 4.9: Pantalla de la gestió d'amics.



#### 4.1.10 Pantalla d'*about*

L'aplicació haurà de tenir la típica pantalla d'*about* on hi haurà la informació de l'aplicació.

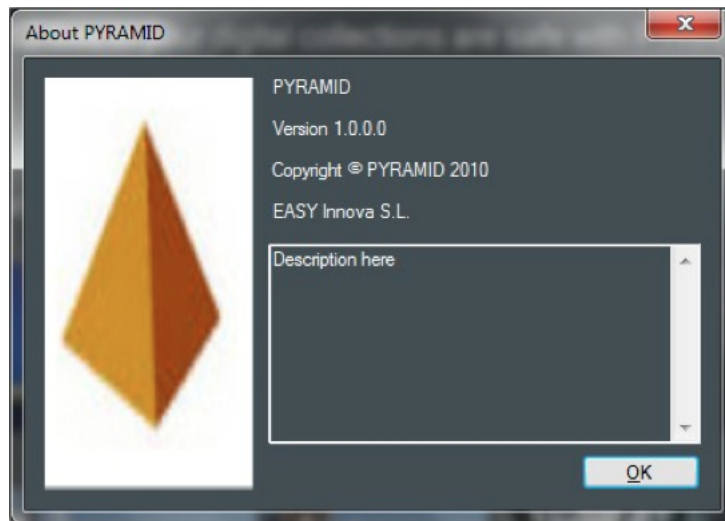


Figura 4.10: Pantalla d'*about*.

## 4.2 Casos d'ús

En aquesta secció de la memòria recollim l'especificació dels casos d'ús que s'han creat o modificat per tal d'afegir els mòduls. S'han obviat els diagrames més simples i s'han afegit els principals.

### 4.2.1 Iniciar sessió

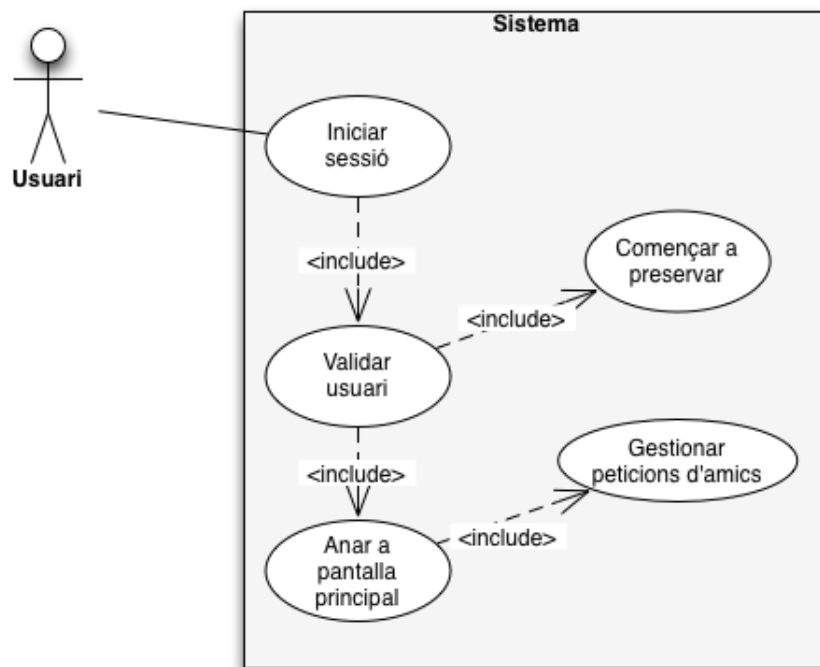


Figura 4.11: Cas d'ús de Inici de sessió.

## 4.2.2 Menú principal

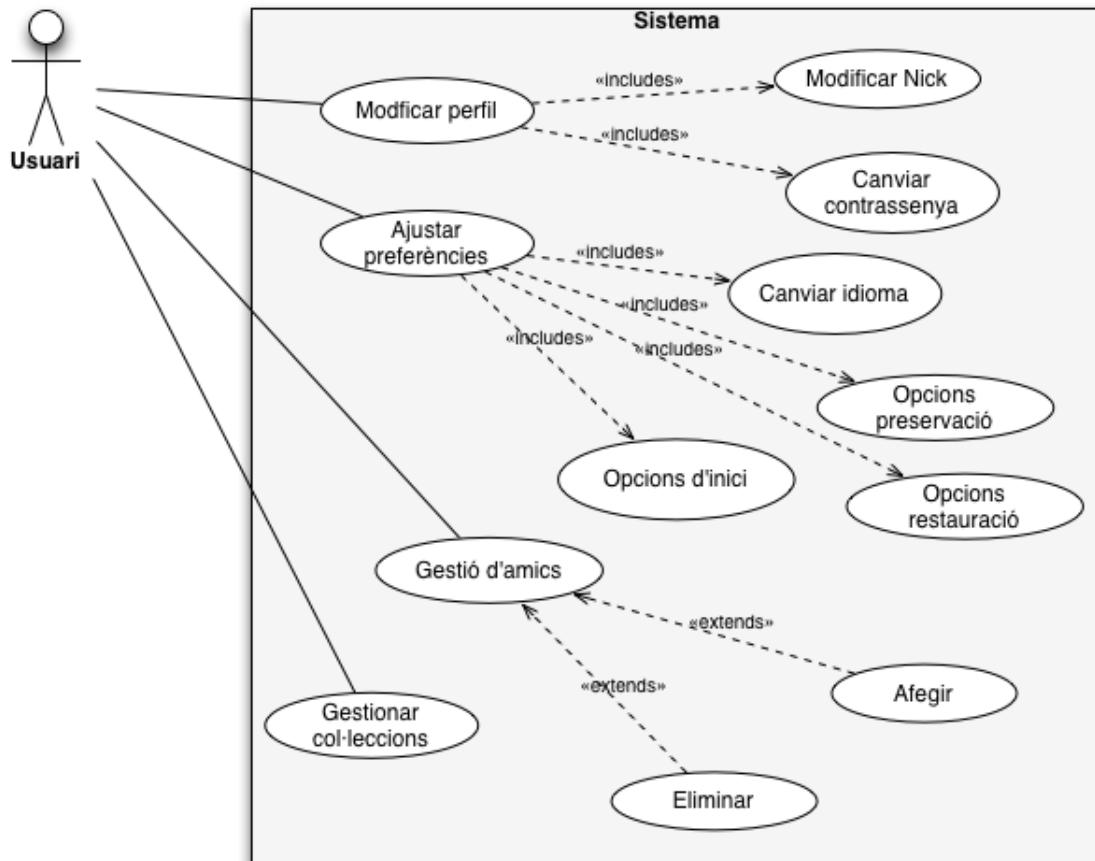


Figura 4.12: Menú principal de l'aplicació.

### 4.2.3 Gestionar col·leccions

S'ha decidit fer aquest cas d'us a part ja que incloure'l al de menú principal dona lloc a un diagrama massa carregat

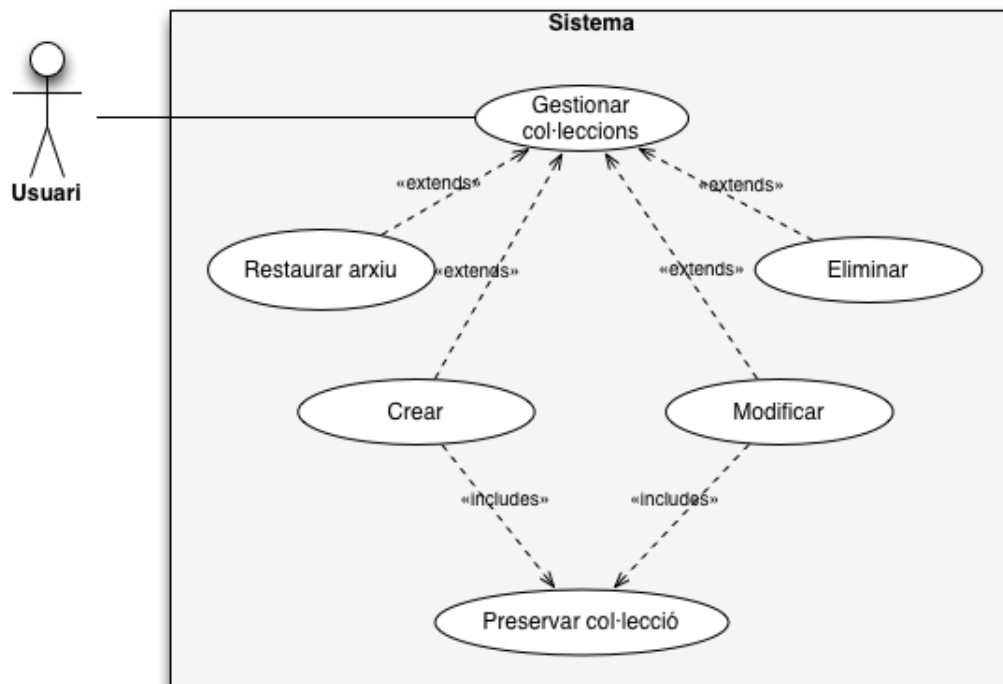


Figura 4.13: Diagrama de cas d'us de la gestió de col·leccions.

## 4.3 Fitxes de cas d'us

### 4.3.1 Iniciar sessió

<b>Nom</b>	Iniciar sessió
<b>Descripció</b>	L'usuari inicia sessió a l'aplicació
<b>Actor</b>	Usuari
<b>Precondició</b>	-
<b>Flux principal</b>	<ol style="list-style-type: none"><li>1. L'actor engega l'aplicació</li><li>2. El sistema mostra una finestra per tal que l'actor introdueixi el seu correu i la seva contrasenya</li><li>3. L'actor introdueix el seu nom d'usuari i la seva contrasenya i prem "Entrar"</li><li>4. El sistema valida les dades introduïdes per l'actor<ol style="list-style-type: none"><li>4.1. El sistema comprova si hi ha peticions d'amistat per acceptar i les mostra per pantalla</li><li>4.2. L'usuari accepta o rebutja les peticions</li></ol></li><li>5. El sistema canvia la plana cap al menú principal de l'actor</li><li>6. El sistema comença a preservar les col·leccions existents</li></ol>
<b>Flux alternatiu</b>	<ol style="list-style-type: none"><li>4. El sistema valida les dades i, si no són correctes, avisa a l'actor que les corregeixi</li></ol>
<b>Postcondició</b>	L'usuari ha iniciat sessió a Pyramid i es troba a la pàgina del seu menú principal

Taula 4.1: Fitxa de cas d'us d'iniciar sessió

### 4.3.2 Crear col·lecció

<b>Nom</b>	Crear col·lecció
<b>Descripció</b>	L'usuari crea una col·lecció i es comença a preservar
<b>Actor</b>	Usuari
<b>Precondició</b>	Usuari <i>loggejat</i>
<b>Flux principal</b>	<ol style="list-style-type: none"><li>1. Clicar a la secció de col·leccions</li><li>2. El sistema mostra les col·leccions ja existents</li><li>3. Es clica al botó de "Nova col·lecció"</li><li>4. L'usuari escull un nom per la col·lecció i el directori on es troben els fitxers que es volen preservar</li><li>5. L'usuari clica al botó de guardar</li><li>6. El sistema afegeix un observer al directori per tal de preservar els nous arxius que s'hi afegeixin</li><li>7. El sistema comença a preservar aquesta col·lecció amb les ja existents</li></ol>
<b>Flux alternatiu</b>	(4 ó 5). L'usuari clica en el botó d'eliminar la col·lecció i aquesta no es guarda
<b>Postcondició</b>	L'usuari ha creat una col·lecció i aquesta ja ha començat a preservar-se

Taula 4.2: Fitxa de cas d'us de crear col·lecció

### 4.3.3 Eliminar col·lecció

<b>Nom</b>	Eliminar col·lecció
<b>Descripció</b>	L'usuari elimina una col·lecció
<b>Actor</b>	Usuari
<b>Precondició</b>	La col·lecció ja existeix
<b>Flux principal</b>	<ol style="list-style-type: none"><li>1. L'usuari clica a la secció de col·leccions</li><li>2. El sistema mostra les col·leccions existents</li><li>3. L'usuari clica en la pestanya de la col·lecció que vol eliminar</li><li>4. L'usuari fa click en el botó d'eliminar</li><li>5. El sistema mostra una finestra de confirmació</li><li>6. El sistema ha eliminat la col·lecció amb els seus arxius</li></ol>
<b>Flux alternatiu</b>	
<b>Postcondició</b>	L'usuari ha eliminat una col·lecció amb els seus arxius

Taula 4.3: Fitxa de cas d'us d'eliminar col·lecció

## 4.4 Diagrames de seqüència

En aquest apartat es veuran uns diagrames de seqüència que il·lustraran el procés d'alguns mòduls, també es podrà observar que, com veurem al següent capítol (Diseny), es segueix un patró *Model View Controller*.

### 4.4.1 Iniciar sessió

Aquest diagrama mostra com es logueja un usuari i el procés de peticions d'amistat.

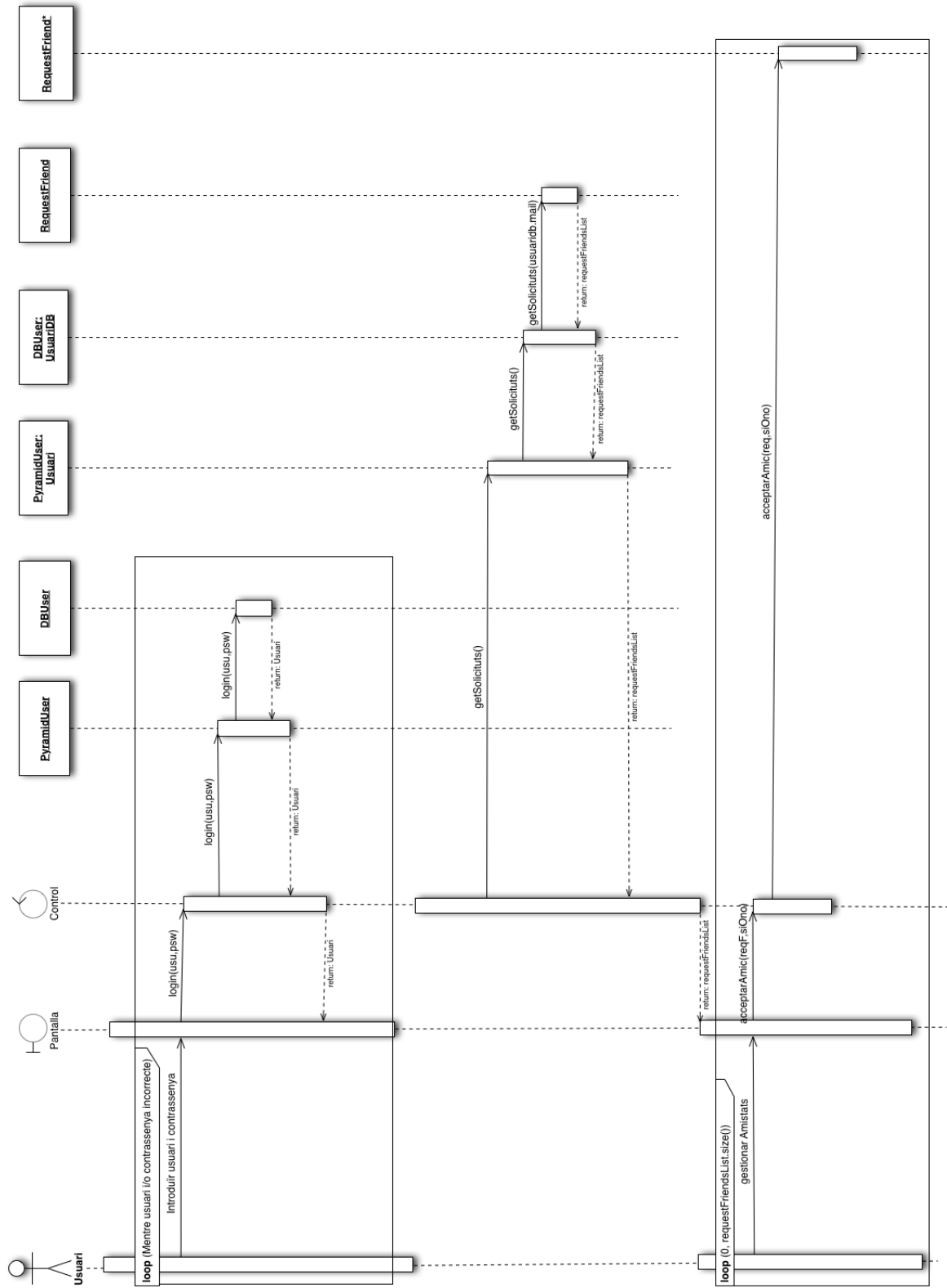


Figura 4.14: Diagrama de seqüència d'iniciar sessió.

## 4.4.2 Crear col·lecció

Aquí es veurà com es crea una col·lecció i per a què es farà us del patró de disseny *Observer*.

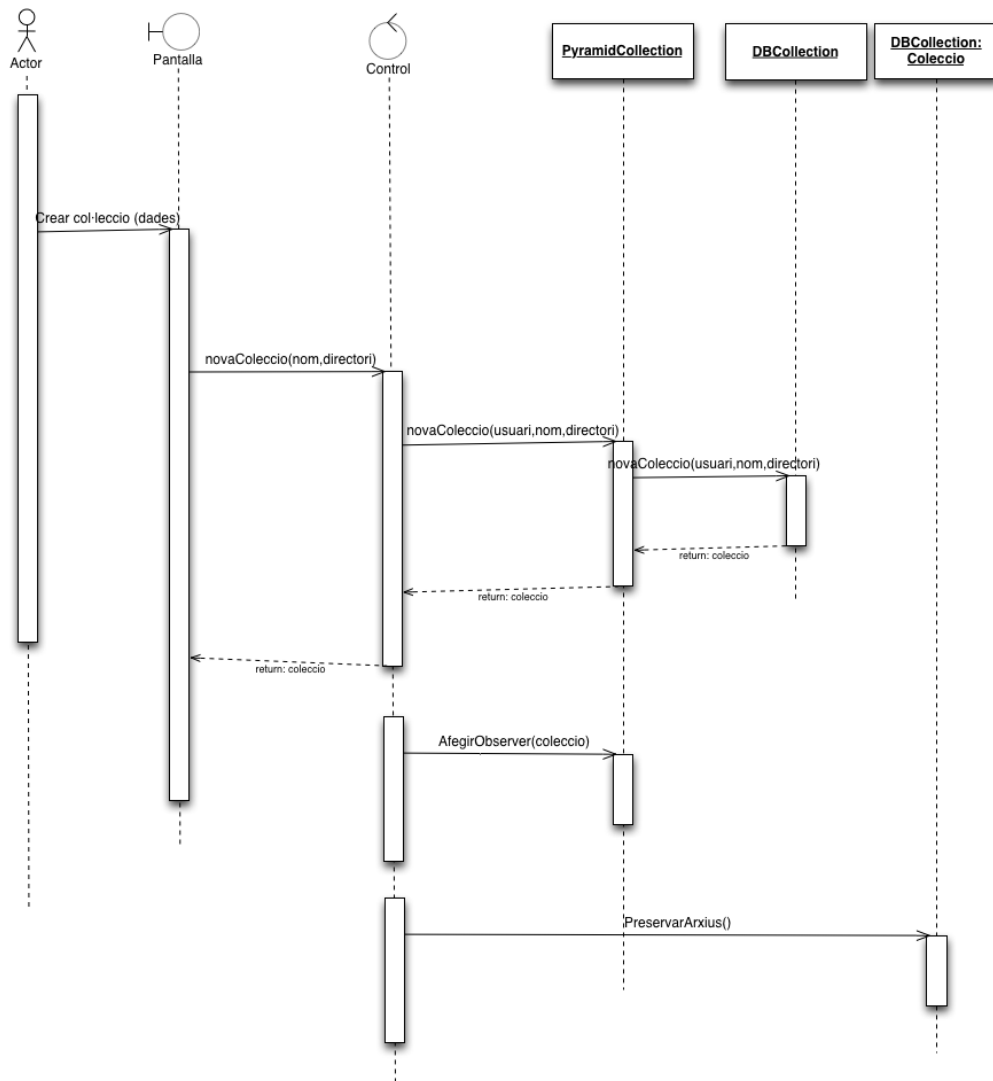


Figura 4.15: Diagrama de seqüència de crear col·lecció.



## 4.5 Diagrama de robustesa

En aquesta secció es generen els diagrames de robustesa a partir dels casos d'ús estudiats prèviament. Aquests diagrames es componen de quatre elements: actors, pantalles, controls i objectes.

Amb aquest diagrama tenim un esquelet de l'aplicació, les funcions i els objectes consultats amb els diferents mètodes d'interacció.

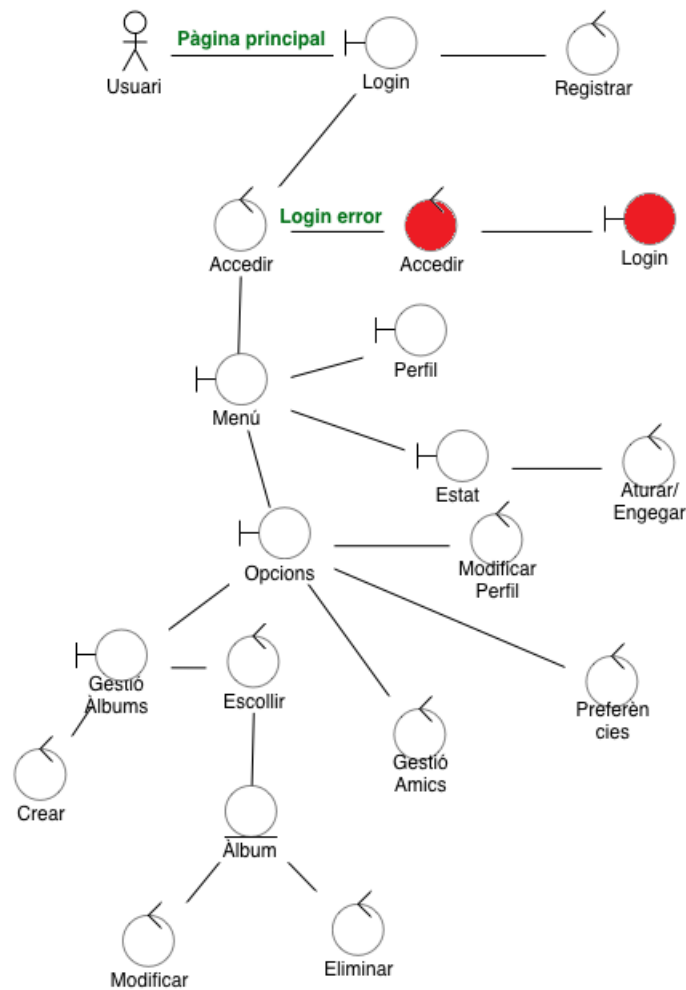


Figura 4.16: Diagrama de seqüència de crear col·lecció.

## 4.6 Diagrama de classes

Per a fer els diagrames de classes hem emprat el Visual Studio 2010 Ultimate, ja que aquest porta integrat un sistema de Ingenieria Inversa que et genera els diagrames a través de les classes del projecte.

Com que és un projecte força gran, amb moltes classes, s'ha generat separant-se en els diferents *namespaces* per, a continuació, analitzar cada mòdul o *namespace* per separat.

### 4.6.1 Disseny en tres capes

A la figura 4.17 es pot veure l'estructura en tres capes (s'explica el patró en el següent capítol (Disseny)), aquestes capes són *Pyramid* (Capa de presentació), *PyramidBusiness* (Capa de negoci) i *PyramidDataLocal* (Capa d'accés a Bdd).

A sota es pot veure un mòdul *PyramidFramework* que l'hem deixat obert per ensenyar que és un mòdul auxiliar per a classes com el log, estadístiques de formats, etc. Finalment, com que la Bdd està situada al servidor al qual hi accedim amb PHP, la part de Bdd s'implementa al mòdul *CookComputing* que són unes llibreries que ens permeten comunicar-nos amb els fitxers PHP que hi ha al servidor i facilitar-nos l'accés a la Bdd.

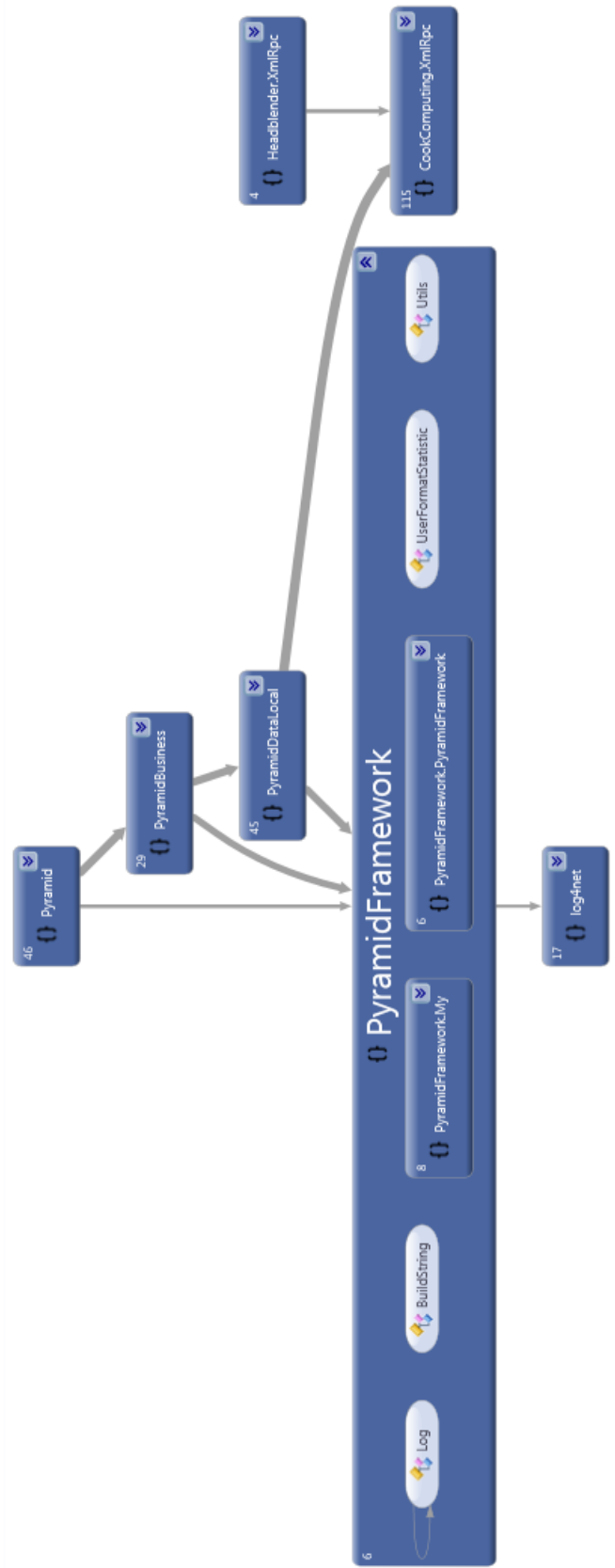


Figura 4.17: Diagrama de classes 3 capes .

## 4.6.2 Capes

### 4.6.2.1 Mòdul Pyramid

Aquest mòdul és la part més alta de les tres capes, la part d'interfície d'usuari, la part amb la que interactuarà l'usuari.

A la figura 4.18 veiem el diagrama de classes d'aquest mòdul. Evidentment, la majoria de les classes d'aquest diagrama es comuniquen amb el seu equivalent de la capa de negoci, per exemple, la classe *PyramidCollectionTab* es comunica amb la classe *PyramidCollection* de la capa de negoci. La classe principal d'aquest mòdul és la *MainWindow*, que és cridada per la *PantallaPrincipal* (la pantalla de benvinguda de l'usuari).

En aquest mòdul s'integra també la part d'auto-actualització de l'aplicació, com es veu sobre el mòdul *Pyramid.My* (aquest mòdul guarda les configuracions i recursos d'aquesta capa).

### 4.6.2.2 Mòdul PyramidBusiness

La figura 4.19 és la pertanyent al diagrama de classes de la capa de negoci.

La classe *PreserverTask* és la classe principal d'aquesta capa i, de fet, de l'aplicació. És el fil d'execució del programa, per on passen totes les consultes i s'encarrega de fer les preservacions. A la part esquerra veiem com s'ha implementat el patró factory, tenim les classes creadores *CreatorDLL* i *CreatorPreserv* que són les encarregades d'instanciar la implementació adequada de l'interfície *IPreserver*. Les classes creadores es cridaran depenent de l'origen del preservador, si és un preservador intern (*PreservBMPtoX*, *SimplePreserv*...) es cridarà el *CreatorPreserv*, en canvi, si és una dll externa donada per un usuari, es cridarà l'altre creador.

### 4.6.2.3 Mòdul PyramidDataLocal

Aquesta és la part més baixa del disseny en tres capes, la capa d'accés a la base de dades (figura 4.20). Tenim un objecte DB per a cada objecte Pyramid de la classe de negoci, cada un d'aquests objectes està relacionats amb la classe *Remote* que porta el seu mateix nom, i a la vegada, aquesta implementa els serveis de les llibreries XMLRPC, per exemple *IDataCollection*. Resumint, es crea una interfície amb els mètodes que es necessiten per accedir a BdD i que inclogui les llibreries XMLRPC del CookComputing (*IDataCollection*), a continuació es crea una classe que implementa aquesta interfície (*DBCcollectionRemote*) que serà la classe nexa entre els objectes "DB" (*DBCcollection*) i la base de dades.

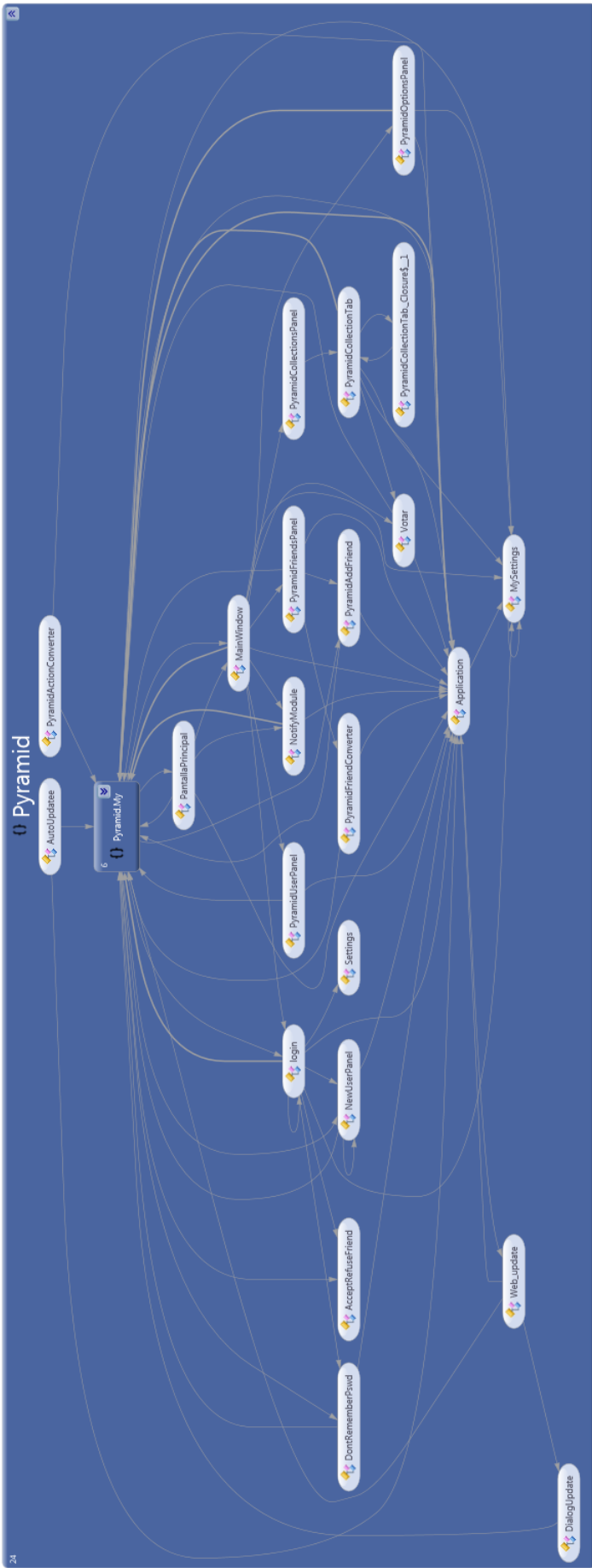


Figura 4.18: Diagrama de classes Pyramid .



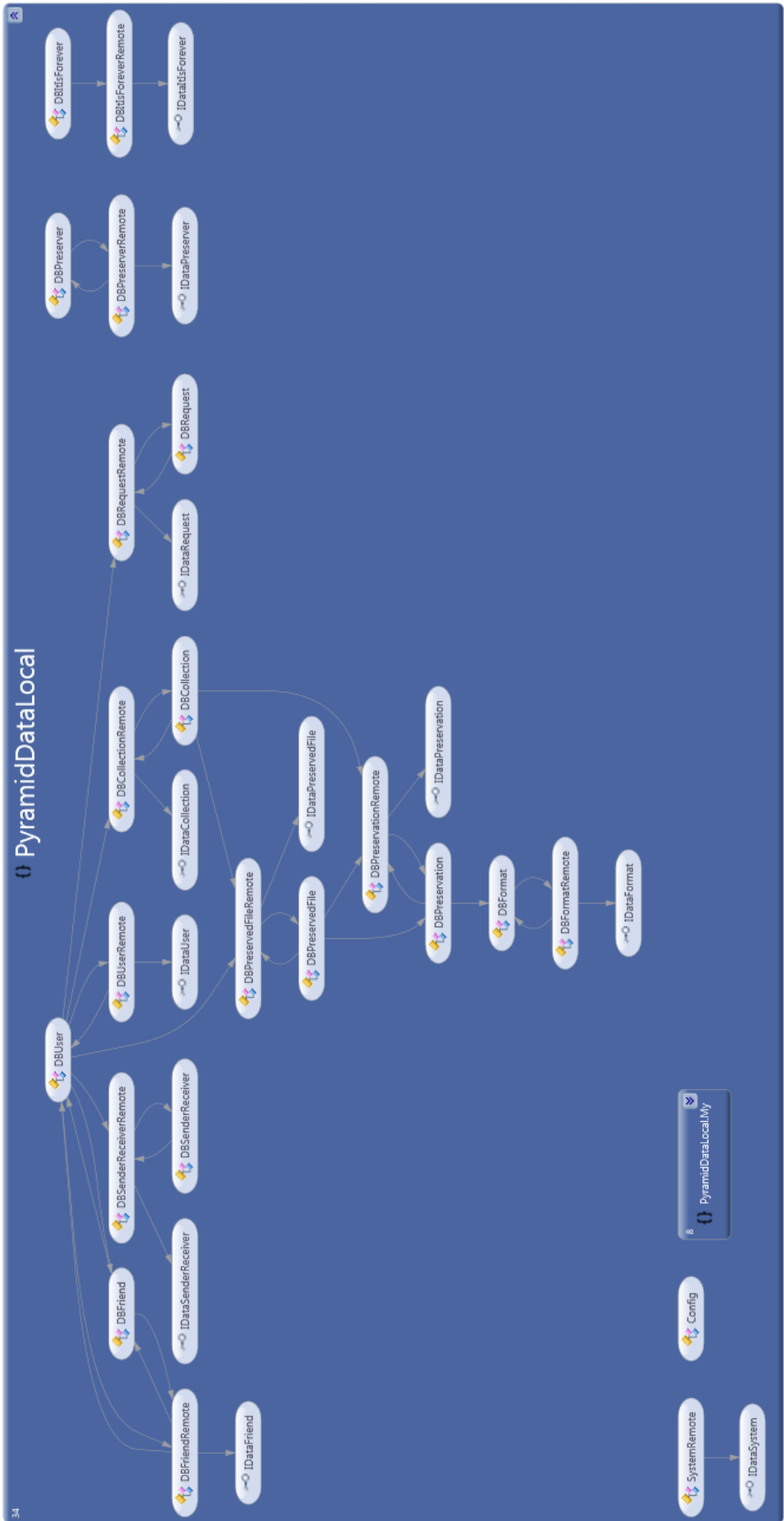


Figura 4.20: Diagrama de classes PyramidDataLocal .

## 4.7 Anàlisi INGENIAS

Aquesta part no entra a la part de Iconix si no que està inclosa dins la part de INGENIAS.

### 4.7.1 Model d'organització

Podem veure que el sistema Pyramid està compost per diferents sistemes, cadascun amb dos agents. Aquests sistemes formen un ecosistema d'agents que s'encarrega d'obtenir nous formats i de preservar el nostre patrimoni en aquests formats.

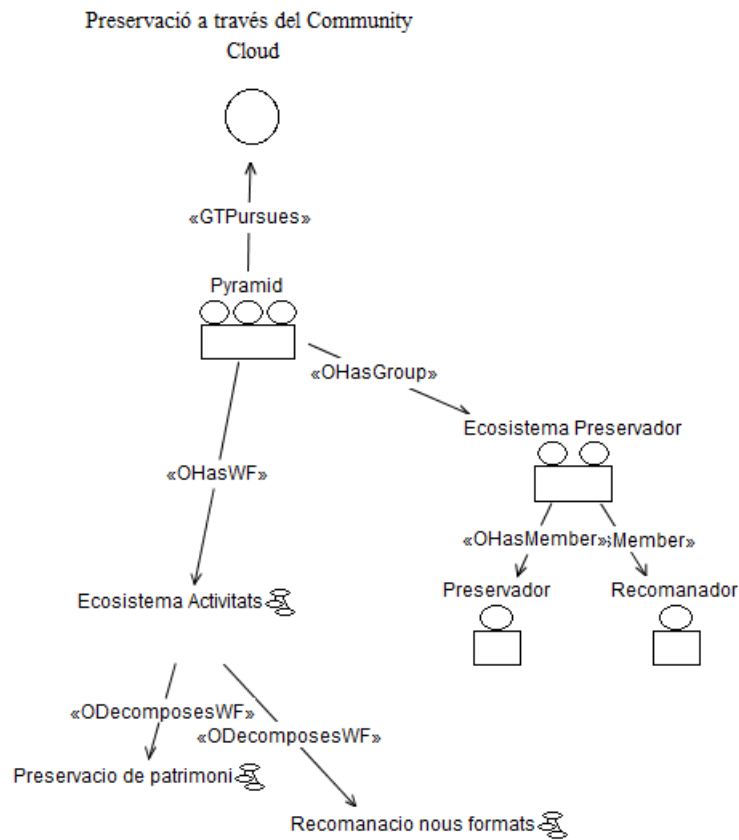


Figura 4.21: Model d'organització



## 4.7.2 Models d'Agents

Per conèixer millor els agents de l'organització, s'elaboren models d'agent. Els objectius dels agents es dedueixen a partir dels objectius de l'organització (les rodones de la part de dalt) i del paper que es preveu que jugaran els agents en aquesta. Els rols (les icones de sota l'agent) sorgeixen associant tasques als objectius. Finalment tenen associats uns processadors i gestors mentals (els cercles de la dreta), que seran del mateix tipus per a tots els agents, i que no hem desenvolupat.

### 4.7.2.1 Agent Preservador

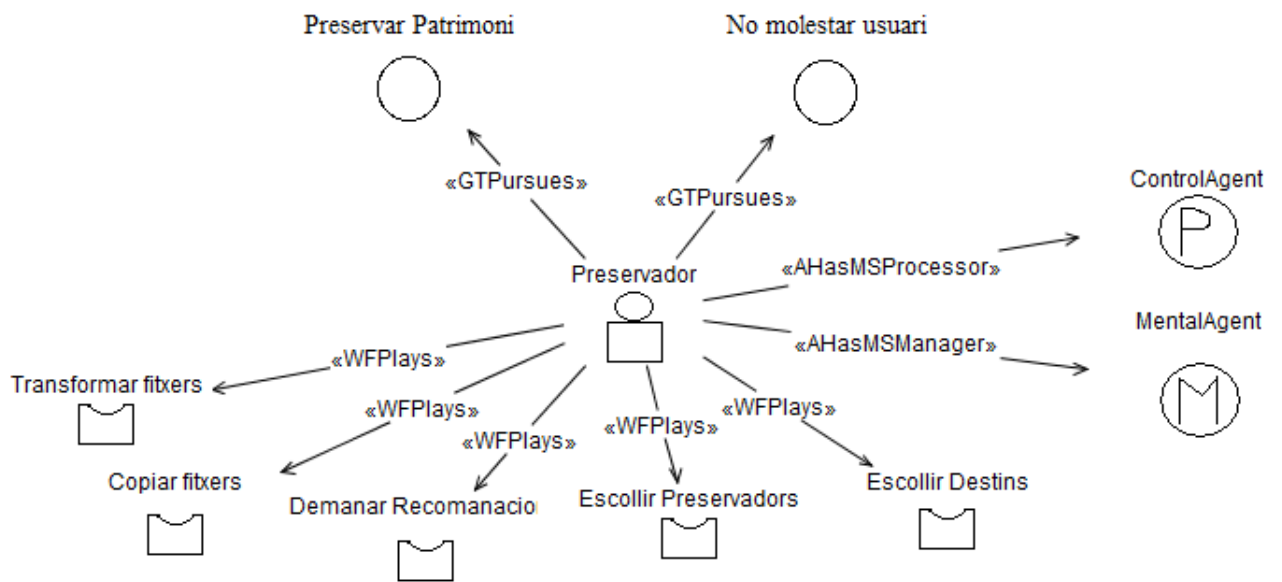


Figura 4.22: Model d'agent Preservador

#### 4.7.2.2 Agent Recomanador

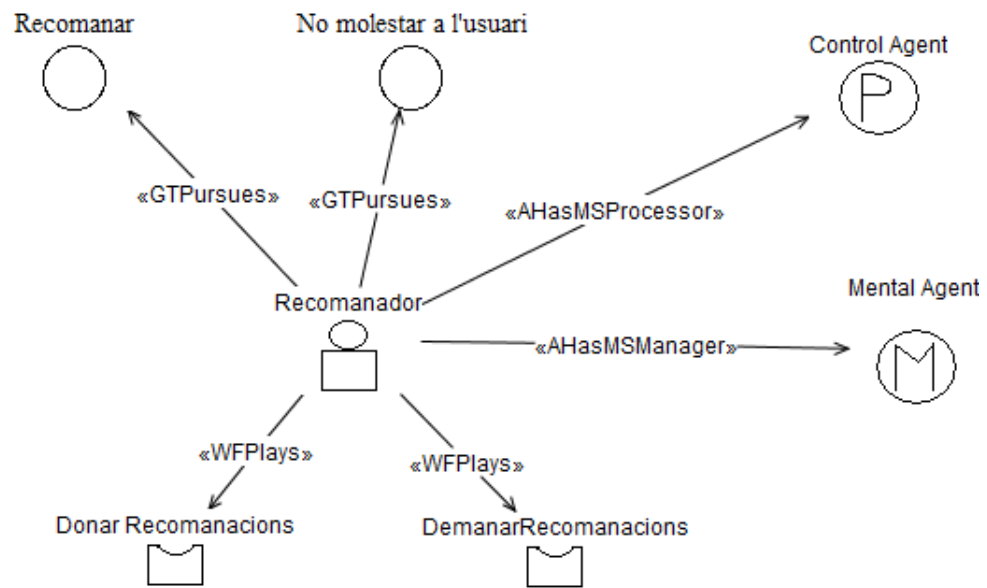


Figura 4.23: Model d'agent Recomanador

### 4.7.3 Models d'interacció

#### 4.7.3.1 Interacció Preservador-Recomanador

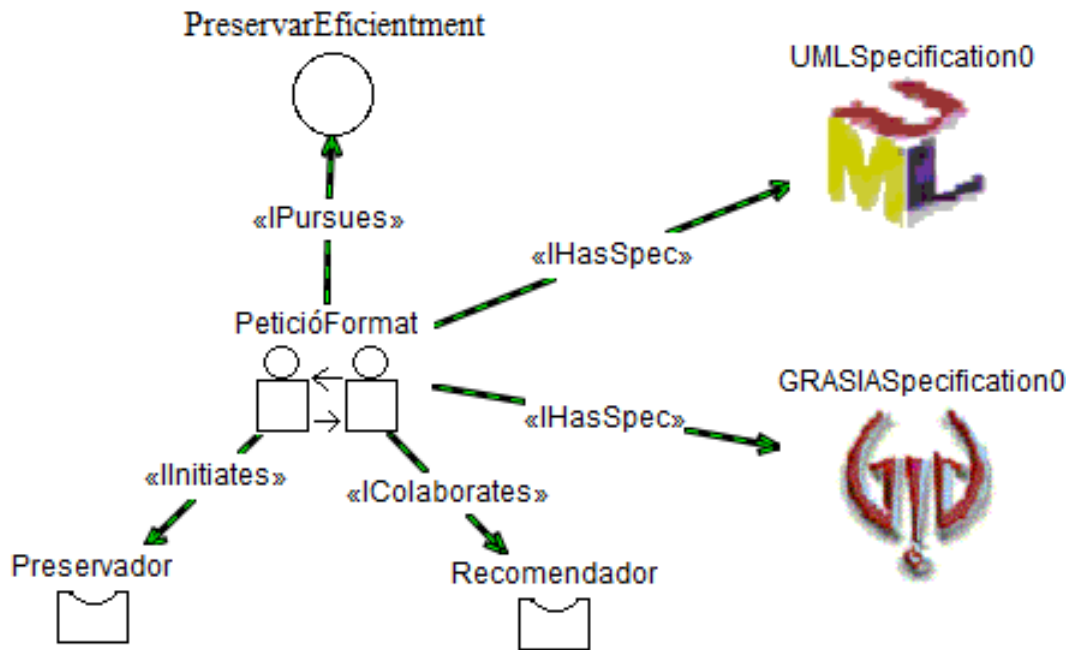


Figura 4.24: Model interacció Preservador-Recomanador

## 4.7.3.2 Interacció Recomanador Local-Recomanador Remot

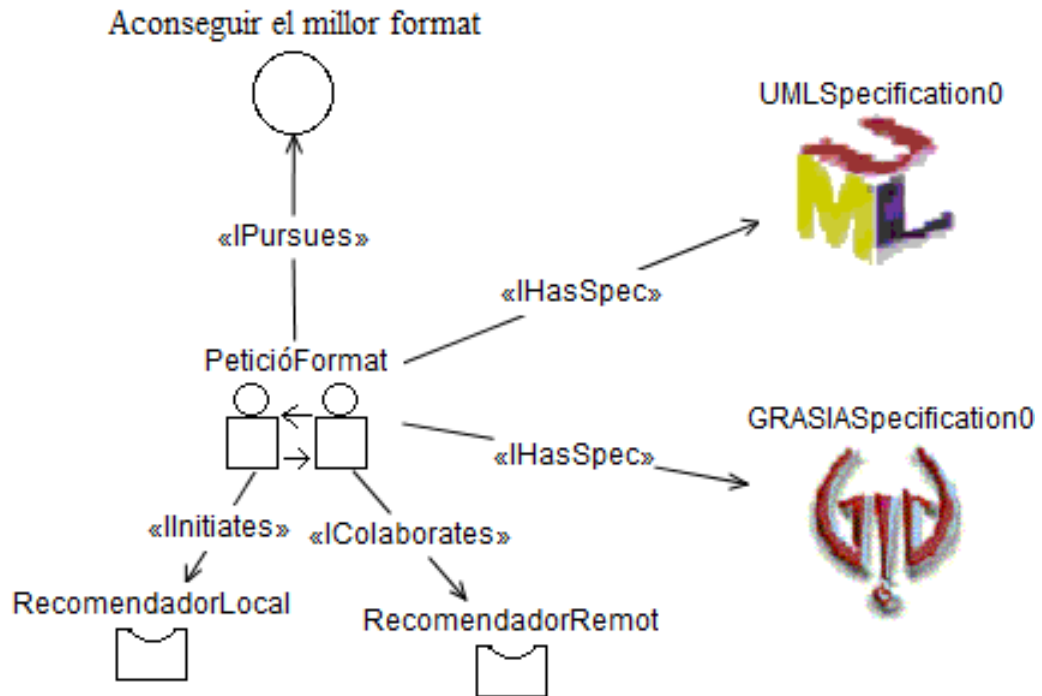


Figura 4.25: Model interacció Recomanador Local-Recomanador Remot

# Capítol 5

## Disseny

Tot i treballar amb la metodologia *Iconix*, que no és precisament de les més pesades, sí trobem una sèrie de patrons de disseny que utilitzem, ja sigui intrínsecament per la tecnologia utilitzada o sigui per qüestions de disseny.

### 5.1 Disseny en tres capes

L'objectiu principal de la programació per capes és la separació de la lògica de negocis de la lògica de disseny; un exemple bàsic d'això consisteix en separar la capa de dades de la capa de presentació a l'usuari.



Figura 5.1: Diagrama d'arquitectura en tres capes.

A continuació es pot veure una imatge de l'IDE amb el codi i es reflexa la separació entre les tres capes.

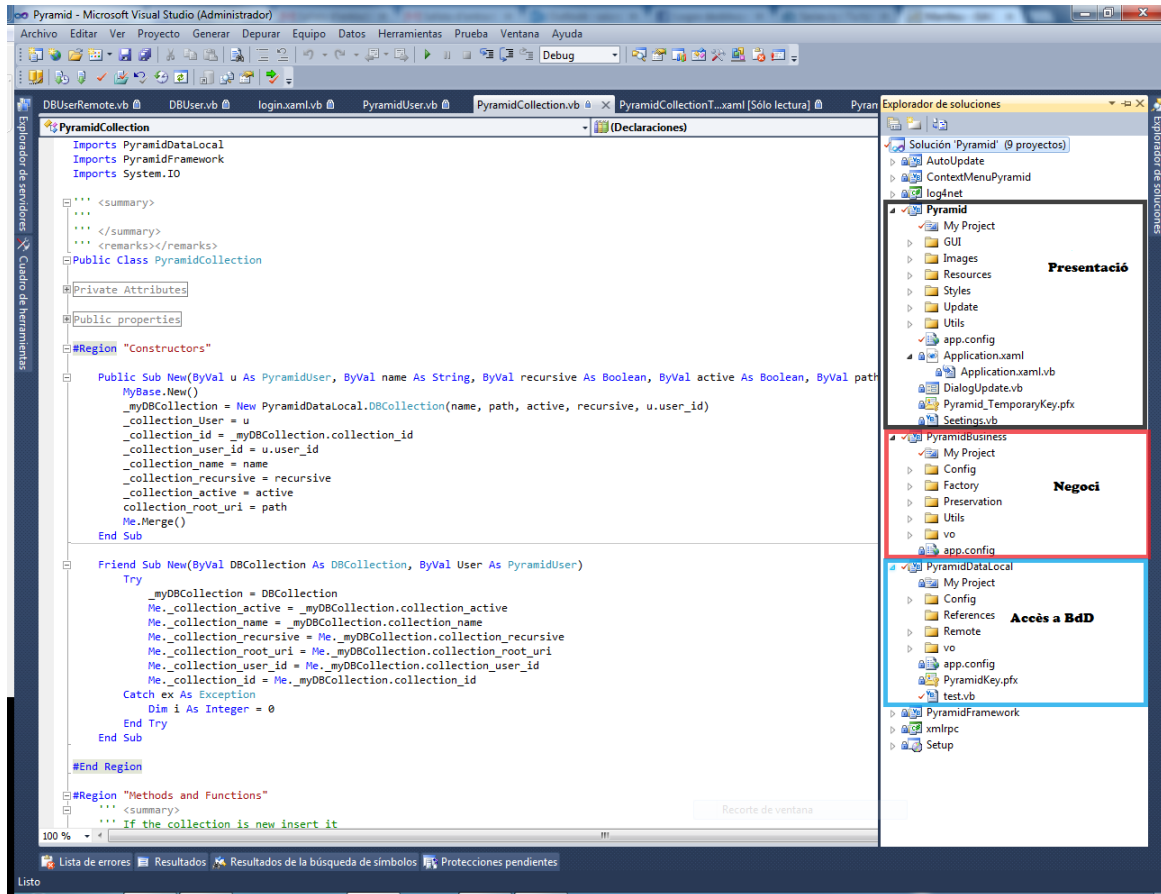


Figura 5.2: Imatge de la jerarquia de capes .

## 5.2 Observer

El patró *Observer* defineix una dependència un-a-molts entre objectes, de tal manera que quan l'objecte canvia d'estat, tots els objectes dependents siguin notificats automàticament. Aquest procediment augmenta la modularitat del llenguatge creant dependències mínimes i evitant bucles d'actualització.

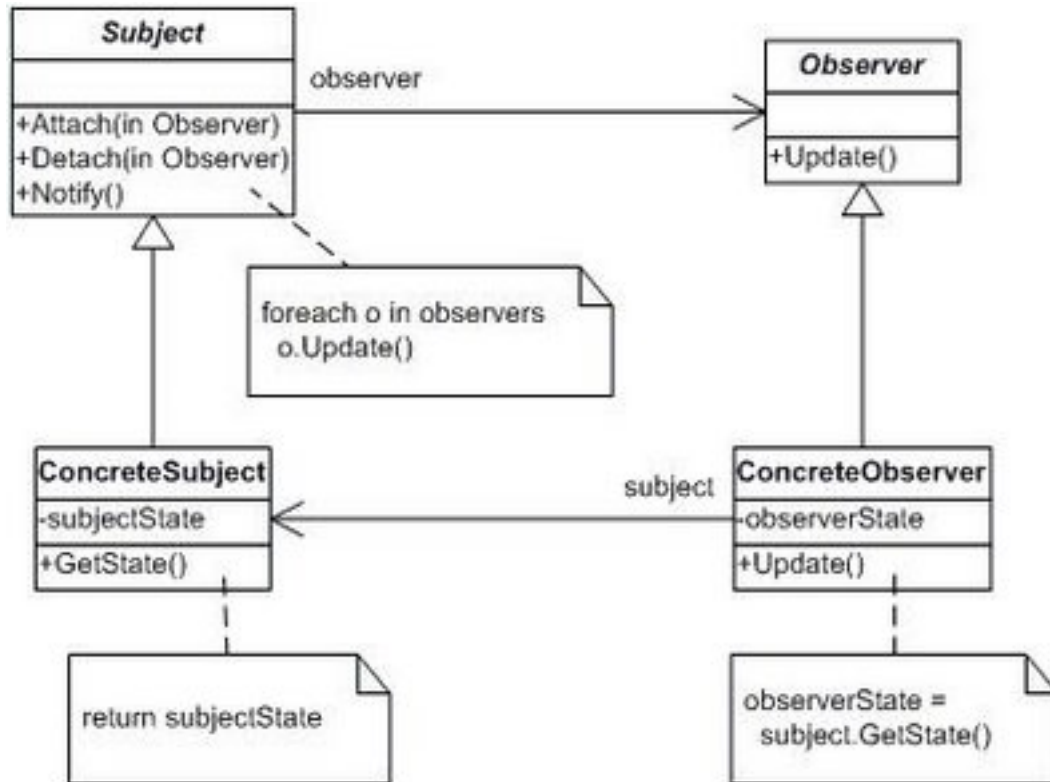


Figura 5.3: Diagrama Observer.

Aquest patró l'hem fet servir per detectar quan s'afegeixen arxius en un directori i començar a preservar-los automàticament.

## 5.3 Visitor

És un patró de comportament, que permet definir una operació sobre objectes d'una jerarquia de classes sense modificar les classes sobre les que opera. Representa una operació que es realitza sobre els elements que conformen l'estructura d'un objecte.

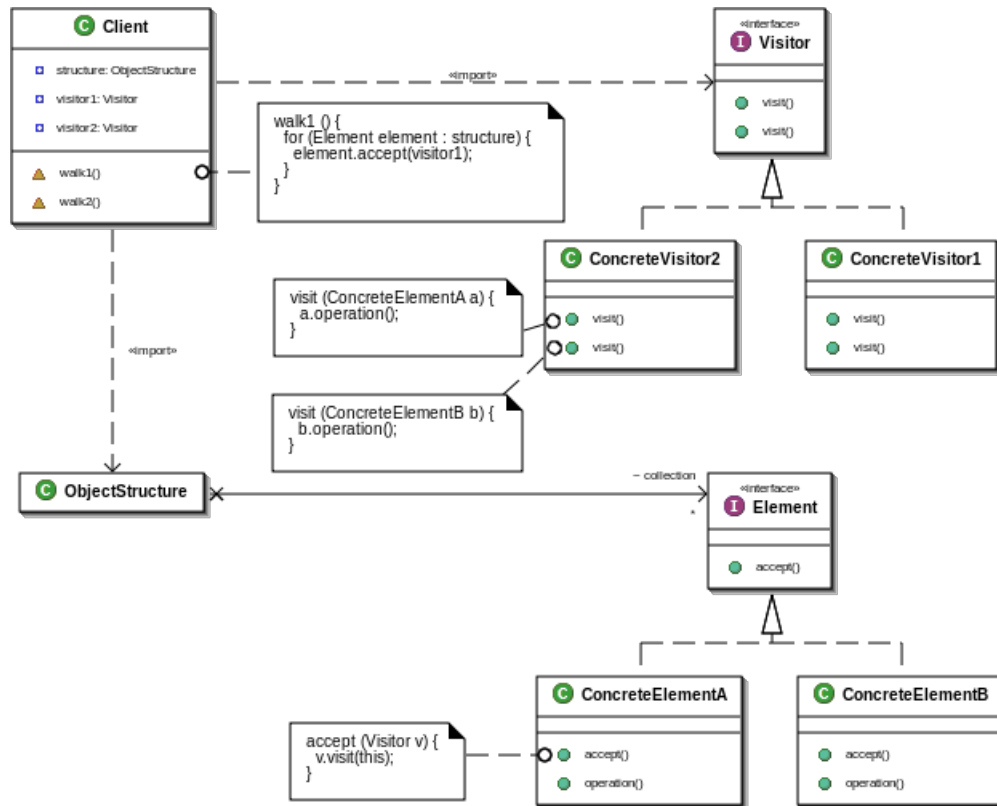


Figura 5.4: Diagrama Visitor.

Aquest patró s'ha emprat per a la modularitat multiidioma, s'encarrega de la càrrega dels textos dels diferents idiomes depenent de l'idioma escollit a les preferències.



## 5.4 Factory Method

El patró *Factory Method* consisteix en utilitzar una classe constructora abstracta amb uns quants mètodes definits i uns altres abstractes: els dedicats a la construcció d'objectes d'un subtipus d'un tipus determinat

Les classes principals en aquest patró són el creador i el producte. El creador necessita crear instàncies de productes, però el tipus concret de producte no ha de ser forçat en les subclasses del creador, perquè les possibles subclasses del creador han de poder especificar subclasses del producte per a utilitzar.

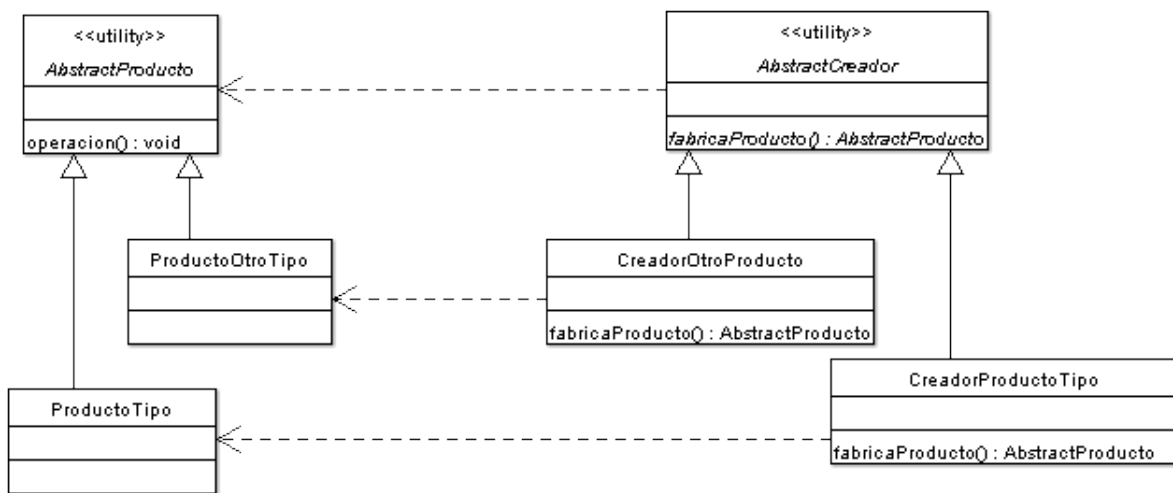


Figura 5.5: Diagrama Factory Method.

La solució consisteix en fer un mètode abstracte (el mètode de la fàbrica) que es defineix en el creador. Aquest mètode abstracte es defineix per a que retorni un producte. Les subclasses del creador poden sobreescrivre aquest mètode per a retornar subclasses apropiades del producte.

Aquest patró serveix en aquest projecte per a facilitar la instanciació dels diferents preservadors depenent del format que es vol preservar.

## 5.5 Singleton

El patró *Singleton* (instància única) està disenyat per a restringir la creació d'objectes pertanyents a una classe o el valor d'un tipus a un únic objecte. La seva intenció consisteix en garantir que una classe només tenguí una instància i proporcionar un punt d'accés global a aquesta.

S'implementa creant un mètode a la nostra classe que crea una instància de l'objecte només si encara no n'existeix alguna.

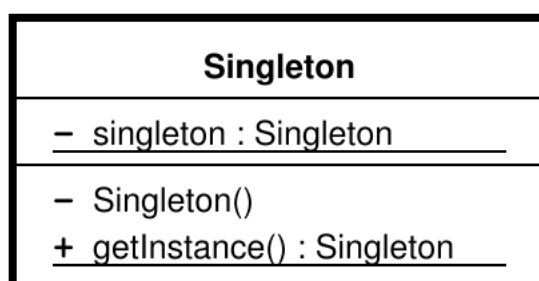


Figura 5.6: Diagrama Singleton.

En aquest projecte es fa servir per la instanciació única del log. Com que és possible que hi hagi diferents fils executant-se, caldrà assegurar que només es crearà un arxiu de log.

# Capítol 6

## Implementació i Proves

En aquest apartat es veurà la implementació dels algorismes principals.

### 6.1 Problemes

Durant el transcurs de la implementació de l'aplicació, ens hem trobat certes dificultats a l'hora d'implementar els agents (a l'annex B mostrarem els típics problemes de la implantació amb agents), així doncs, ens hem adonat que per aquesta primera versió pre-alpha, era més adient l'us d'un *thread* principal i consultes amb la base de dades deixant, així, la implementació d'agents per a futures millores.

### 6.2 Preservadors

A continuació s'explicarà com s'ha implementat tot el tema de preservadors. Com s'ha explicat a l'apartat de disseny, per a la instanciació dels preservadors s'ha fet servir un patró factory i ara es reflectirà en els subapartats següents.

#### 6.2.1 Interfície

Cada preservador haurà d'implementar una interfície que hem anomenat IPreserver. Aquesta haurà de contenir tres mètodes:

- **Preserv:** Mètode que rebrà per paràmetre l'arxiu a preservar, l'acció a fer (copia, transformació, local, remot, amic) i el format destí. Aquest mètode

retorna un String de la forma *String:: string:host or name PC +”|”+ string: URL where the file is saved +”|”+ int: action*

- **PreservByte:** Mètode que rebrà per paràmetre el mateix que el mètode anterior, però retornarà un array de *Byte* amb la imatge convertida a *Bytes*.
- **GetValue:** Mètode que no rep paràmetres i retorna els valors que pot transformar aquest preservador en forma de diccionari: *Dictionary(format origen, llista de formats destí)*, per exemple *[JPG,[BMP,GIF,PNG,TIF]]*.

Aquesta interfície es podrà consultar i descarregar per tal de que els usuaris puguin fer els seus propis preservadors i penjar-los al servidor, un cop penjats, nosaltres comprovarem que sigui correcte i el posarem a actiu per a que el programa el pugui descarregar i fer-lo servir com a dll.

A continuació podem veure com queda la interfície en VB.Net

```

Public Interface IPreserver
    *** <summary>
    *** Preserv Method than convert Format X in Format Y
    *** </summary>
    *** <returns>String:: string:host or name PC +"|" + string: URL where the file is saved +"|" int: action
    *** action=
    *** LOCAL_COPY = 1
    *** REMOTE_COPY = 2
    *** FRIEND_COPY = 3
    *** LOCAL_TRANSFORMATION = 4
    *** REMOTE_TRANSFORMATION = 5
    *** FRIEND_TRANSFORMATION = 6
    *** </returns>
    *** <remarks></remarks>
    Function Preserv(ByVal file As PyramidFile, ByVal action As PyramidAction, ByVal dest As String) As String

    *** <summary>
    *** Preserv Method than convert Format X in Format Y
    *** </summary>
    *** <returns>Byte() than carry the ImageFile to preserv in main program
    *** </returns>
    *** <remarks></remarks>
    Function PreservByte(ByVal file As PyramidFile, ByVal action As PyramidAction, ByVal formatdesti As String) As Byte()

    *** <summary>
    *** Save the compatibility of preserver
    *** </summary>
    *** <returns>
    *** Dictionary(origin format, list of destiny formats)
    *** For example:
    *** [JPG,[BMP,GIF,PNG,TIF]]
    *** </returns>
    *** <remarks></remarks>
    Function GetValues() As Dictionary(Of String, ArrayList)
End Interface

```

Figura 6.1: Interfície del Preservador.

En la següent imatge es pot veure com s'han general els diccionaris dels preservadors locals.

## 6.2.2 Instanciar preservador

Com s'ha explicat anteriorment, s'ha implementat un patró factory per a la instanciació dels preservadors, es diferencien "creadors/instanciadors" de dos tipus, un per als locals, que s'han implementat en el desenvolupament de l'aplicatiu i un per a les dll (*Dynamic-Link Library*) que penjaran els usuaris en el servidor:

- **CreatorPreserv:** El creador dels preservadors locals, els que tenen extensió **.vb**
- **CreatorDLL:** El creadors dels preservadors externs, que estan penjats al

```

Dim compatibility As New Dictionary(Of String, ArrayList)
Dim compatibility2 As New Dictionary(Of String, ArrayList)
Dim llista As New ArrayList()
Dim llista2 As New ArrayList()
llista.Add("PreservBMPtoX")
compatibility.Add("Tiff", llista)
compatibility.Add("jpeg", llista)
compatibility.Add("jpg", llista)
compatibility.Add("png", llista)
compatibility.Add("gif", llista)
llista.Add("SimplePreserverImp")
compatibility.Add("bmp", llista)
compt.Add("bmp", compatibility)

llista2.Add("PreservMTIFtoX")
compatibility2.Add("Tiff", llista)
compatibility2.Add("jpeg", llista)
compatibility2.Add("jpg", llista)
compatibility2.Add("png", llista)
compatibility2.Add("gif", llista)
llista2.Add("SimplePreserverImp")
compatibility2.Add("mtiff", llista2)
compt.Add("mtiff", compatibility2)

```

Figura 6.2: Diccionaris dels preservadors locals.

servidor amb extensió **.dll**

El mètode principal d'aquesta classe es diu CreatePreserv i té dues sobrecàrregues, una amb el nom del preservador i l'altra amb un número a l'atzar i la llista de preservadors. El que fa és recuperar el nom del preservador i l'instància fent un casting amb IPreserv.

```

Dim assembly As Assembly = assembly.GetExecutingAssembly()
Try
    ConvertObject = AppDomain.CurrentDomain.CreateInstance(assembly.FullName, PathObj)
    obj = DirectCast(ConvertObject.Unwrap(), IPreserver)
    Return obj
Catch ex As Exception
    Console.WriteLine("[CreatorPreserver.CreatePreserv]]" + ex.Message)
    log.Err("[CreatorPreserver.CreatePreserv]", ex)
    Return Nothing
End Try

```

Figura 6.3: Codi del creador de preservadors.

## 6.2.3 Ampliació de preservadors

En el thread principal, hi haurà un comprovador de si hi ha nous preservadors comparant els que tenim guardats a BdD amb la llista del servidor.

En el cas de que hi hagi un de nou, se'l descarrega, inspecciona el diccionari i afegeix en el diccionari general els formats inici i destí que pot transformar aquest preservador.

```
''' <summary>
''' Downloads the possible updates.
''' </summary>
''' <returns>Returns a list filled with the names of available preservadors</returns>
''' <remarks></remarks>
Public Function GetLibraries() As List(Of String)
    Dim llista As New List(Of String)
    Try
        Dim l As List(Of PyramidDataLocal.DBPreserver) = PyramidDataLocal.DBPreserverRemote.getPreservers()
        For Each preservador As PyramidDataLocal.DBPreserver In l
            If preservador.Preserver_Type = 1 Then
                Try
                    If Not IO.Directory.Exists(Config.LocalPreserversFolder) Then
                        IO.Directory.CreateDirectory(Config.LocalPreserversFolder)
                    End If
                    llista.Add(preservador.Preserver_Name)
                    My.Computer.Network.DownloadFile(PyramidDataLocal.Config.preservers_url & preservador.Preserver_Name, Config.LocalPreserversFolder & preservador.Preserver_Name.ToString)

                    Catch ex As Exception
                        Console.WriteLine("[Getdll.GetLibraries DownloadFile]" + ex.StackTrace)
                        log.Err("[Getdll.GetLibraries DownloadFile]", ex)
                    End Try
                End Try
                'llista.Add(preservador.Preserver_Name)
            Next
        Catch ex As Exception
            Console.WriteLine("[Getdll.GetLibraries]" + ex.Message)
            log.Err("[Getdll.GetLibraries]", ex)
            llista = New List(Of String)
        End Try
    Return llista
End Function
```

Figura 6.4: Descàrrega del preservador.

### 6.2.3.1 Exemple

Per a poder crear els propis preservadors, es crea un projecte nou de tipus "Biblioteca de classes" i com a referències hi poses el nostre Pyramid.Business.

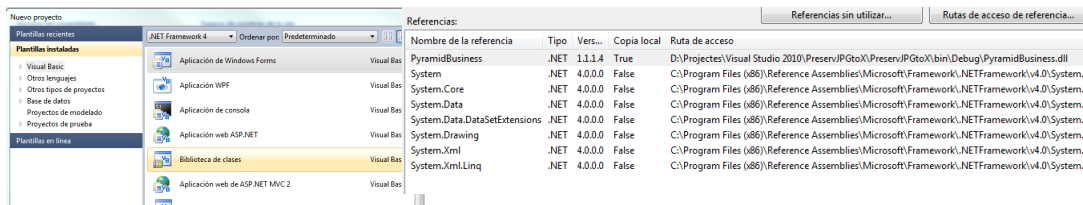


Figura 6.5: Nou projecte llibreria.

Figura 6.6: Referències.

A continuació crees una classe anomenada Preservador que implementi iPreserver, aquesta, obligatòriament, haurà de contenir els mètodes esmentats a l'apartat de la interfície, ets lliure d'implementar tants mètodes auxiliar com desitgis.

```
Public Class Preservador
    Implements IPreserver

    Public Function Preserv(ByVal myfile As PyramidBusiness.PyramidFile, ByVal action As PyramidBusiness.PyramidAction, ByVal dest As String) As String Implements PyramidBusiness.IPrese
        ''' <summary>
        ''' Funcio per a convertir de jpg a qualsevol altre format    '''
        ''' </summary>
        ''' <param name="FullPath"></param>
        ''' <param name="dest"></param>
        ''' <returns></returns>
        ''' <remarks></remarks>
    Private Function Convertir(ByVal FullPath As String, ByVal dest As String) As String ...

    Public Function GetValues() As Dictionary(Of String, ArrayList) Implements IPreserver.GetValues
        Dim compatibility As New Dictionary(Of String, ArrayList)
        Dim llista As New ArrayList()
        llista.Add("tiff")
        llista.Add("jpeg")
        llista.Add("jpg")
        llista.Add("exif")
        llista.Add("bmp")
        llista.Add("icon")
        llista.Add("wmf")
        llista.Add("bmp")
        llista.Add("png")
        llista.Add("MemoryBmp")
        llista.Add("gif")
        llista.Add("Emf")
        compatibility.Add("JPG", llista)
        Return compatibility
    End Function

    Public Function PreservByte(ByVal file As PyramidFile, ByVal action As PyramidAction, ByVal formatdesti As String) As Byte() Implements IPreserver.PreservByte ...

    Private Function ConvertirBytes(ByVal BMPFullPath As String, ByVal dest As String) As Byte() ...

End Class
```

Figura 6.7: Nou preservador.



Un cop fet això, generes el projecte, busques l'arxiu .dll generat i el penjes al servidor [http://84.88.145.20/pyramid\\_updates/Preservers/](http://84.88.145.20/pyramid_updates/Preservers/) via ftp. Un cop penjat, comprovarem que funcioni correctament i l'aplicació s'encarregarà de carregar-lo.

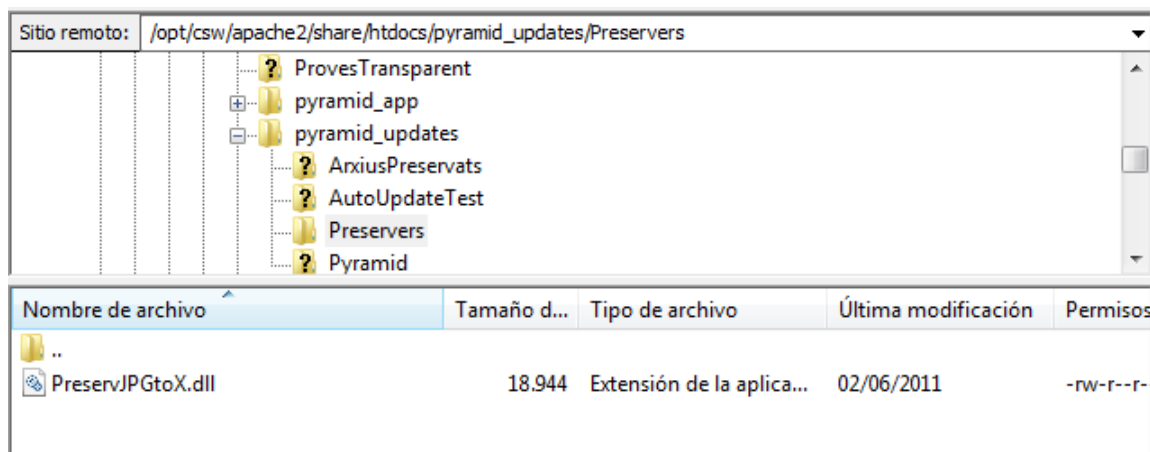


Figura 6.8: Dll penjada al servidor.

Un cop l'equip tècnic comprovi que el nou preservador sigui operatiu, és a dir, que funcioni correctament i faci les transformacions de manera òptima, etc. El posarem en actiu per a que els *Pyramid* operatius en els ordinadors dels usuaris se'l descarreguin automàticament.

Els diccionaris de cada preservador, un cop descarregats, s'afegeixen a un diccionari general del fil principal on de cada format origen, et diu els seus destins i quins preservadors poden transformar, per exemple: [origen][destins| [ preservadors ] ] [.jpg][.bmp | [jpgToX], .png| [jpgToX] , jpg | [jpgToX, preservadorCopia] ].

A l'exemple tenim que si tenim una foto .JPG, aquesta es pot transformar a diferents formats amb jpgToX, però per a fer una còpia de .jpg a .jpg tenim dos preservadors diferents.

## 6.3 Preservar

En aquesta secció s'explicarà tot el procés de preservació que segueix un arxiu des de que s'afegeix la col·lecció, queda preservat en algun destí i fem la restauració d'aquest.

### 6.3.1 Afegir col·lecció

Per afegir col·leccions, el que es fa en realitat és afegir el directori on es troben els arxius que es volen preservar.

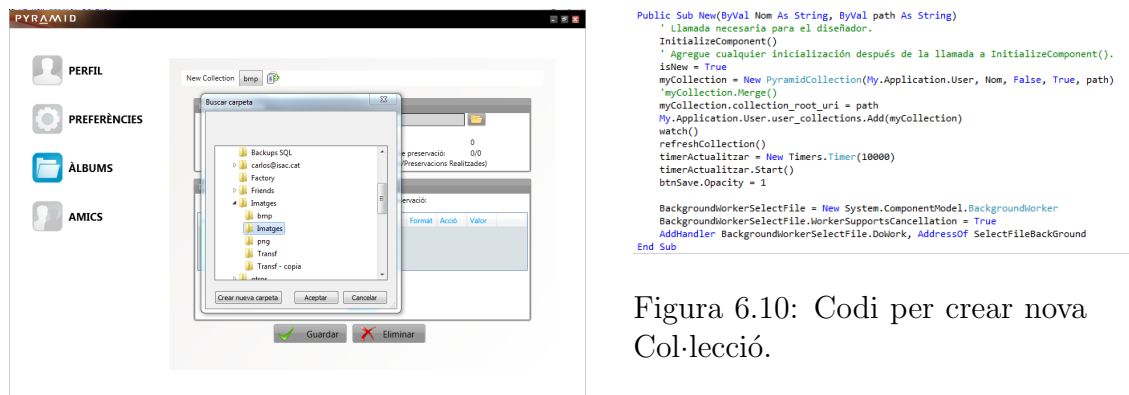


Figura 6.9: Afegir Col·lecció.

Un cop s'afegeix la col·lecció, s'hi col·loca un *observador* per a que si s'hi afegeix un nou arxiu directament al directori, l'aplicació ho detecti i l'afegeixi a la col·lecció, llavors s'agafen tots els arxius que hi ha a la col·lecció i s'afegeixen a la llista de preservacions pendents.

El *refreshCollection* que es pot veure al codi, comprova els arxius que s'han anat preservant i, si estan preservats afegeix una imatge de *thumbs up* per a que l'usuari vegi que realment és així.

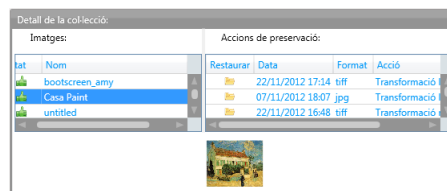


Figura 6.11: Imatge preservada.

## 6.3.2 Fil principal de preservació

Mentre s'explica el procés de preservació, apareixerà al codi que s'ecollirà el mètode i destí de la preservació, aquests els descriurem al final de la secció.

El procés de preservació tracta d'un *thread* separat que, de forma resumida, va agafant els fitxers un a un i els va preservant segons el que cregui convenient.

```
Dim myPyramidFile As PyramidFile = user.getUnpreservedFile() 'S'agafa el primer arxiu no preservat
Dim urlSrc As New Uri(myPyramidFile.file_uri, UriKind.Absolute)

Dim nomFitxer As String = TreureEspais(urlSrc.AbsolutePath) 'Canviem "%20" per " "

Dim ext As String = GetFileNameWithoutExtension(TreureEspais(urlSrc.AbsolutePath))
Dim extensio As String = nomFitxer.Substring(ext.Length + 1)

Dim destiEsc As String = EscollirPreserver(extensio) 'Escollim el preservador

Dim dest As String
dest = ""

Dim ChosenPreserv As Integer = escollirLRA() 'Escollim entre local, remot i amic
'1 = local
'2 = remot
'3 = amic

dest = Preservar(ChosenPreserv, destiEsc, myPyramidFile, extensio) 'Preservem el nostre arxiu
preserver = user.getPreserverByName(ChosedPreserv.ToString)
GuardarBdD(dest, myPyramidFile) 'Guardem la nostra preservació a BdD si s'ha preservat bé
```

Figura 6.12: Thread principal.

Si seguim el codi línia a línia, en primera instància s'agafa el primer arxiu no preservat i a continuació s'agafa la *URI* (*Uniform Resource Identifier*) de l'arxiu, és a dir, la ruta d'on es troba físicament, llavors es tracta el nom, ja que per defecte els espais es substitueixen (exemple: un arxiu que es digui "soc jo.jpg" ens retornaria "soc%20jo.jpg").

Llavors, separem el nom de l'extensió del fitxer i escollim el preservador segons el format origen de l'arxiu. Si tenim amics, es farà un càlcul sobre els arxius que han puntuat millor els nostres amics, en cas contrari s'ecollirà un format destí a l'atzar d'un preservador escollit a l'atzar.

Llavors escollim un dels destins físics on guardar l'arxiu preservat. Finalment, preservem l'arxiu passant com a paràmetre el destí on es guardarà, el format en que es guardarà, l'arxiu i el format origen.

A continuació podem veure el codi de la funció *Preservar*.

```

Select Case LocalRemoteFriend 'Mirem quin tipus de preservació (local/remota/amic) hem escollit
Case 1 'local
Dim size As Integer = 0
If Directory.Exists(DestinationFolder & "/" & myPyramidFile.file_Collection.collection_user_id.ToString) Then 'Agafem la mida del directori
For Each f As FileInfo In New DirectoryInfo(DestinationFolder).GetFiles("", SearchOption.AllDirectories)
size += f.Length
Next
End If
If (size / 1000000) < maxLocalSizeFolder Then 'Si sobrepassem el límit de capacitat (definida al panell d'opcions), no preservem
If (String.Compare(destiesc.ToLower, origen.ToLower) = 0) Then 'Mirem quina acció farem
accio = PyramidAction.ACTION.LOCAL_COPY 'còpia local"
Else
accio = PyramidAction.ACTION.LOCAL_TRANSFORMATION 'transformació local
End If
dest = Local(myPyramidFile, destiesc.ToLower, accio) ' destiesc és el format destí escollit
End If
Case 2 'remot
If (String.Compare(destiesc.ToLower, origen.ToLower) = 0) Then 'Mirem quina acció farem
accio = PyramidAction.ACTION.REMOTE_COPY 'còpia remota
Else
accio = PyramidAction.ACTION.REMOTE_TRANSFORMATION 'transformació remota
End If
dest = Me.Upload(myPyramidFile, destiesc.ToLower, accio) 'destiesc és el format destí escollit
Case 3 'amic
If (String.Compare(destiesc.ToLower, origen.ToLower) = 0) Then 'Mirem quina acció farem
accio = PyramidAction.ACTION.FRIEND_COPY 'còpia amic
Else
accio = PyramidAction.ACTION.FRIEND_TRANSFORMATION 'transformació amic
End If
dest = Me.SendToFriend(myPyramidFile, destiesc.ToLower, accio) 'destiesc és el format destí escollit
End Select
Return dest

```

Figura 6.13: Funció Preservar.

Com es pot comprovar al codi, tenim un mètode diferent per a cada destí. En els subapartats següents s'ensenyarà què fa cada mètode. Recuperem el preservador usat i guardem a base de dades conforme s'ha preservat bé.

### 6.3.2.1 Destins de preservació

Com s'ha explicat anteriorment, disposem de tres destins diferents per a preservar. Aquests destins són:

#### Local

Com el propi nom indica, aquest destí et fa còpies locals, en el directori que escollis en el menú d'opcions.

Si ets una persona que modifica constantment una col·lecció i sense voler has borrat un arxiu. Aquesta és la forma més ràpida de recuperar-lo.

A continuació es podrà observar el codi de preservació d'arxius en local:

```

Dim size As Long = PyramidFramework.Utils.DirectorySize(DestinationFolder & "/" & File.file_Collection.collection_user_id.ToString, True) / 1000000
If (size) < maxLocalSizeFolder Then
    Dim desti As String = DestinationFolder & "/" & File.file_Collection.collection_user_id.ToString & "/"
    & File.file_Collection.collection_id.ToString & "/" & File.file_id.ToString
    Dim Arxiu As Byte() = ChosedPreserv.PreservByte(File, New PyramidAction(), formatDest)

    If (Arxiu.Length > 0) Then
        Dim oFileStream As System.IO.FileStream
        If Not IO.Directory.Exists(desti) Then
            IO.Directory.CreateDirectory(desti)
        End If
        desti = (desti & "/" & GetFileNameWithoutExtension(File.file_uri) & "." & formatDest)
        oFileStream = New System.IO.FileStream(desti, System.IO.FileMode.Create)
        oFileStream.Write(Arxiu, 0, Arxiu.Length)
        oFileStream.Close()
        Return (System.Environment.MachineName + "|" + desti + "|" + accio)
    Else
        Return "-1"
    End If
Else
    Return "-1"
End If

```

Figura 6.14: Preservació local.

Els tres mètodes per a guardar en diferents destins no varien gaire, la idea és la mateixa per tots tres:

- Recuperar adreça destí.
- Tractar l'arxiu amb un dels mètodes (copia o transformació).
- Guardar l'arxiu tractat amb la nova extensió al destí.
- Retornar al fil principal si hi ha hagut èxit i donar la informació.

Si el problema és que s'ha mort l'ordinador o s'ha formatat el disc dur, llavors hi ha les següents dues formes de preservar arxius.

## Remot

Aquest mètode et preserva els arxius en un servidor remot per tal de que, si es dona un dels casos que s'acaba d'explicar, es formati l'ordinador o aquest es mori, des d'un altre ordinador amb pyramid puguis iniciar sessió i recuperar els arxius que s'hagin preservat al servidor.

Els passos a seguir són els mateixos que en local, però aquest cop l'adreça destí és un servidor i, per tan, s'ha de fer un *upload* de l'arxiu cap al servidor.

```

Dim urlSrc As New Uri(myFile.file_uri, UriKind.Absolute)
Dim Arxiu As Byte() = ChosedPreserv.PreservByte(myFile, New PyramidAction(), destiesc) 'Fem la transformació que pertoca
If (Arxiu.Length > 0) Then

    Dim dest As String = myFile.file_uri 'Fem un split i agafem la ubicació de l'arxiu
    Dim nomArxiu As String = GetFileNameWithoutExtension(dest) & "." & destiesc

    'Dim Uri As String = Config.preserved_folder + "/" + myFile.file_Collection.Collection_User.user_id.ToString

    result = PyramidDataLocal.DBPreservationRemote.uploadPreservation(nomArxiu, myFile.file_id, myFile.file_Collection.collection_id,
myFile.file_Collection.collection_user_id, Convert.ToBase64String(Arxiu))
    'ftp.Upload(My.Computer.FileSystem.SpecialDirectories.Temp & "\" & nomArxiu, nomArxiu)
    If Not result.Equals("-1") Then
        result = PyramidDataLocal.Config.server + "|" + result + "|" + accio
    End If
End If

```

Figura 6.15: Preservació remota.

## Amic

Aquesta és la opció per si fallen les altres dues o per si s'ocupa el màxim d'espai per als altres dos destins.

```

Dim Arxiu As Byte() = ChosedPreserv.PreservByte(myFile, New PyramidAction(), formatdesti) 'Fem la transformació que pertoca
Dim nomArxiu As String = GetFileNameWithoutExtension(myFile.file_uri) & "." & formatdesti
Dim dest As String = myFile.file_Collection.collection_user_id & "/" & myFile.file_Collection.collection_id & "/" & myFile.file_id & "/"
Dim arxiuString As String = Convert.ToBase64String(Arxiu) 'Convertim el byteArray en String per enviar-lo
'escollim, entre els amics connectats, a qui li enviem
Dim amics As New List(Of PyramidFriend)
amics = user.getFriendsAccepted
Dim Rand As Integer = amics.Count
Dim Random As New Random(Now.Millisecond)
If (Rand > 0) Then
    Dim ChosenFriend As Integer = Random.Next(1, Rand)

    'Procedim a enviar l'arxiu
    Dim sender As New PyramidSenderReceiver(amics(ChosenFriend - 1).Friend_friend_id, user.user_id, dest, nomArxiu, arxiuString)
    Dim id As Integer = sender.Send()
    If id > 0 Then
        Return amics(ChosenFriend - 1).Friend_Mail + "|" + dest + "/" + GetFileName(nomArxiu) + "|" + accio.ToString + "|" + id.ToString
    Else
        Return "-1"
    End If
Else
    Return "-1"
End If

```

Figura 6.16: Preservació amiga.

Els passos són els mateixos que a les altres dues opcions, però aquí el destí és el pc de l'amic, així doncs, es comprova que l'amic tingui el pyramid encés i, si és així, es descarrega i es marca l'arxiu com a preservar; en cas contrari, el fitxer es queda a servidor a l'espera de que l'amic obri el pyramid. Mentrestant l'arxiu original es pot seguir preservant de les altres formes.

Els arxius no es preservaran només d'una única forma, s'aniran preservant regularment pels tres destins i en diferents formats, a l'apartat de Restaurar es veurà que hi ha una llista amb les preservacions fetes d'un mateix arxiu.

### 6.3.2.2 Mètodes de preservació

Hi ha dos mètodes que engloven les preservacions, hi ha la còpia, el format origen i el destí són iguals i la transformació, canvien l'extensió i format de l'arxiu destí respecte l'origen.

#### Copia

El primer preservador que es va implementar va ser un preservador copia, que agafava l'arxiu entrant i el clonava, per tal de tenir exactament el mateix arxiu sense modificar.

```
Try
  If myFile.file_Collection Is Nothing Then
    Return "-1"
  End If
  Dim urlSrc As New Uri(myFile.file_uri, UriKind.Absolute)
  Dim ff As New IO.FileInfo(myFile.file_uri)
  Dim dest2 As String = dest + "/" + myFile.file_Collection.collection_user_id.ToString + "/" +
myFile.file_Collection.collection_id.ToString + "/" + myFile.file_id.ToString
  If Not IO.Directory.Exists(dest2) Then
    IO.Directory.CreateDirectory(dest2)
  End If
  IO.File.Copy(myFile.file_uri, dest2 + "/" + ff.Name, True)
  Return (System.Environment.MachineName + "|" + dest2 + "/" + ff.Name + "|" + "1")
Catch ex As Exception
  Console.WriteLine("[SimplePreserverImpPreserv] " + ex.Message)
  log.Err("[SimplePreserverImpPreserv]", ex)
  Return "-1"
End Try
```

Figura 6.17: Preservació copia.

Com es veu a la imatge, aprofitem les llibreries de vb.Net per a fer una copia de l'origen al destí. Aquest preservador s'afegeix al diccionari general esmpre que aparegui un nou preservador.

És a dir, si encara no tenim cap preservador de .doc a .docx, per exemple, si n'apareix un, al diccionari general si podria afegir origen: .doc destins: .docx:doc, .pdf, .odt... als destins hi afegim .doc (si no hi és) i posem com a preservador aquest preservador còpia (com s'ha explicat al final de la secció d'ampliació de preservadors).

**Transformació** El segon mètode de preservació és la transformació, és a dir, que el format destí i l'origen siguin diferents, per exemple, passar d'un arxiu .bmp a un arxiu .jpg.

```

Try
    If myFile.file_Collection Is Nothing Then
        Return "-1"
    End If
    Dim urlSrc As New Uri(myFile.file_uri, UriKind.Absolute)
    Dim ext As String
    ext = urlSrc.AbsolutePath.Substring(urlSrc.AbsolutePath.LastIndexOf(".") + 1).ToLower()
    If String.CompareOrdinal(ext, "BMP") = 0 Or String.CompareOrdinal(ext(1), "bmp") = 0 Then
        Dim ff As New IO.FileInfo(myFile.file_uri)
        dest = dest + "/" + myFile.file_Collection.collection_user_id.ToString + "/"
        + myFile.file_Collection.collection_id.ToString + "/" + myFile.file_id.ToString
        dest = Me.Convertir(urlSrc.OriginalString, dest)
    Else
        Return "-1"
    End If
    Return System.Environment.MachineName + "|" + dest + "|4"
Catch ex As Exception
    Console.WriteLine("[PreservBMPToX.Preserv]" + ex.Message)
    log.Err("[PreservBMPToX.Preserv]", ex)
    Return "-1"
End Try

```

Figura 6.18: Preservació transformació.

El mètode principal no varia gaire del mètode còpia, però podem observar que el *File.Copy* no hi és i s'ha substituït per un mètode *Convertir*.

```

Private Function Convertir(ByVal BMPFullPath As String, ByVal dest As String) As String
    Try
        Dim dest2 As String = dest
        Dim imgFormat As ImageFormat
        Dim form As Integer = seleccioFromDest1("bmp")
        Dim objBmp As New Bitmap(BMPFullPath)
        Select Case form
            Case 1
                imgFormat = ImageFormat.Jpeg
                If Not IO.Directory.Exists(dest2) Then
                    IO.Directory.CreateDirectory(dest2)
                End If
                dest2 = dest2 + "/" + GetFileNameWithoutExtension(BMPFullPath)
            Case 2
                imgFormat = ImageFormat.Emf
                If Not IO.Directory.Exists(dest2) Then
                    IO.Directory.CreateDirectory(dest2)
                End If
                dest2 = dest2 + "/" + GetFileNameWithoutExtension(BMPFullPath)
            .
            .
            .
            Case Else
                imgFormat = ImageFormat.Bmp
                If Not IO.Directory.Exists(dest2) Then
                    IO.Directory.CreateDirectory(dest2)
                End If
                dest2 = dest2 + "/" + GetFileNameWithoutExtension(BMPFullPath)
        End Select
        dest2 &= "." & imgFormat.ToString
        objBmp.Save(dest2, imgFormat)
        Return dest2
    Catch ex As Exception
        Console.WriteLine("[PreservBMPToX.Convertir]" + ex.Message)
        log.Err("[PreservBMPToX.Convertir]", ex)
        Return "-1"
    End Try
End Function

```

Figura 6.19: Funció convertir.



Depenent del preservador podrem transformar a uns formats o altres, aquí veiem que l'origen és BMP i els format en que es poden convertir (s'ha acotat per no allargar la memòria amb codi redundant).

S'escull el format destí i es transforma en aquest, a continuació es guarda la imatge a memòria i es retorna.

A més, a cada preservador tenim un mètode per a transformar a un array de bytes directament per tal de facilitar l'enviament (aquest és el mètode que es fa servir actualment).

```
Private Function ConvertirBytes(ByVal BMPFullPath As String, ByVal dest As String) As Byte()
    Try
        Dim objBmp As New Bitmap(BMPFullPath)
        Dim ms As New MemoryStream()
        If (String.CompareOrdinal(dest, ".jpg") = 0 Or String.CompareOrdinal(dest, ".jpeg") = 0) Then
            objBmp.Save(ms, ImageFormat.Jpeg)
        ElseIf (String.CompareOrdinal(dest, ".emf") = 0) Then
            objBmp.Save(ms, ImageFormat.Emf)
        ElseIf (String.CompareOrdinal(dest, ".exif") = 0) Then
            objBmp.Save(ms, ImageFormat.Exif)
        ElseIf (String.CompareOrdinal(dest, ".gif") = 0) Then
            objBmp.Save(ms, ImageFormat.Gif)
        ElseIf (String.CompareOrdinal(dest, ".icon") = 0) Then
            objBmp.Save(ms, ImageFormat.Icon)
        ElseIf (String.CompareOrdinal(dest, ".memorybmp") = 0) Then
            objBmp.Save(ms, ImageFormat.MemoryBmp)
        ElseIf (String.CompareOrdinal(dest, ".png") = 0) Then
            objBmp.Save(ms, ImageFormat.Png)
        ElseIf (String.CompareOrdinal(dest, ".tiff") = 0) Then
            objBmp.Save(ms, ImageFormat.Tiff)
        ElseIf (String.CompareOrdinal(dest, ".wmf") = 0) Then
            objBmp.Save(ms, ImageFormat.Wmf)
        Else
            objBmp.Save(ms, ImageFormat.Bmp)
        End If
        Return ms.ToArray()
    Catch ex As Exception
        Console.WriteLine("[PreservBMPtoX.Convertir]" + ex.Message)
        log.Err("[PreservBMPtoX.Convertir]", ex)
        Return New Byte() {}
    End Try
End Function
```





Figura 6.20: Funció convertirBytes.

## 6.3.3 Restaurar

La finalitat de preservar els arxius és que més endavant els puguis recuperar. En aquest apartat s'explica com fer-ho.

### 6.3.3.1 Versió usuari

A la imatge següent podem veure com a la finestra d'opcions, a l'apartat d'àlbums, ens apareixen els àlbums en diferents pestanyes i cada àlbum te una llista d'arxius.

-  **PERFIL**
-  **PREFERÈNCIES**
-  **ÀLBUMS**
-  **AMICS**

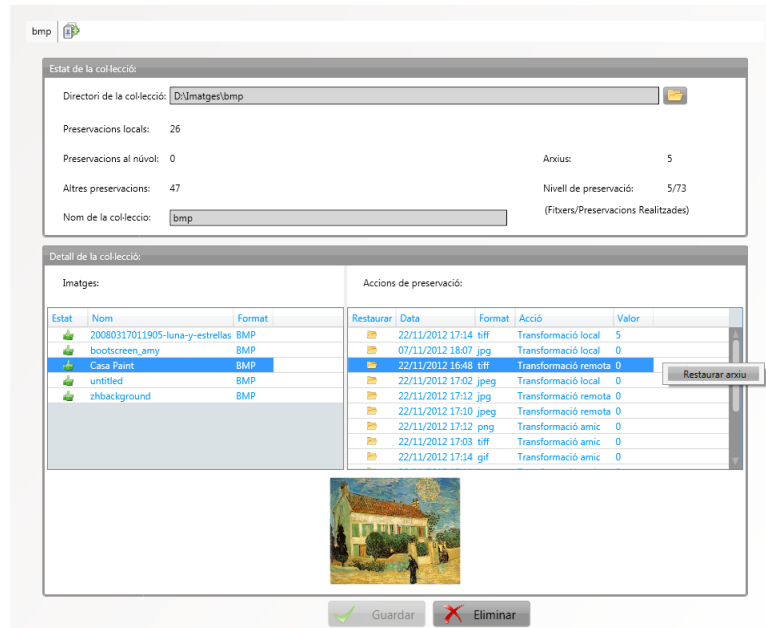


Figura 6.21: Restaurar arxiu.

En clicar, per exemple, a Casa Paint, que és un arxiu BMP, ens surt la llista de preservacions que s'ha fet. Volem restaurar una transformació remota que s'ha fet en format tiff, fem click dret i donem a restaurar.

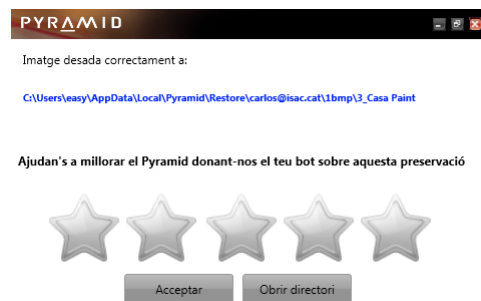


Figura 6.22: Imatge restaurada.

Apareix la imatge anterior on podem obrir el directori on s'ha desat la imatge clicant a l'enllaç o el botó d'obrir directori. Votem com ens sembla la preservació per tal de que els amics sàpiguen quin preservador va millor segons el criteri de qui puntua.

### 6.3.3.2 Versió programador

Com que tenim tres destins possibles per preservar, també tenim tres formes possibles de restaurar. Hi ha una part de codi que és comuna per totes tres: Aquesta part

```
Dim elements As PyramidPreservation
Dim puntuacio As Integer = 0
elements = dGrdActions.CurrentItem 'Agafem l'arxiu a restaurar
Dim origen As String() = Split(elements.Preservation_destination, "/") 'Desglosem l'adreca origen
Dim desti As String = My.Settings.RestoreFolder + "\" + _
    elements.Preservation_file.file_Collection.Collection_User.user_email + "\" + _
    elements.Preservation_file.file_Collection.collection_id.ToString + "-" + _
    elements.Preservation_file.file_Collection.collection_name + "\" + _
    elements.Preservation_file.file_id.ToString + "-" + _
    GetFileNameWithoutExtension(elements.Preservation_file.file_name) 'Especifiquem el directori on ho guardarem

Dim servidor As String = elements.Preservation_host 'Agafem el primer element per veure si es local o remot

Dim accio As Integer = elements.Preservation_action_id 'Agafem l'acció de l'element per veure si es local o remot
If Not Directory.Exists(desti) Then 'si el directori restaurador no existeix, es crea
    Directory.CreateDirectory(desti)
End If
```

Figura 6.23: Part comuna de restauració.

recupera la informació de l'arxiu (nom, usuari, col·lecció...), a la variable servidor hi guardem si és local, remot o amic i creem la carpeta on es guarden les restauracions si encara no s'ha creat.

A continuació es veurà la part específica per a cada destí.

#### Remot

```
If elements.Preservation_action_id = PyramidAction.ACTION.REMOTE_COPY Or elements.Preservation_action_id = PyramidAction.ACTION.REMOTE_TRANSFORMATION Then
    Dim data As Byte() = elements.getPreservationData()
    If data.Length > 1 Then
        If File.Exists(desti + "\" + elements.Preservation_name) Then
            File.Delete(desti + "\" + elements.Preservation_name)
        End If
        Dim f As FileStream = File.OpenWrite(desti + "\" + elements.Preservation_name)
        f.Write(data, 0, data.Length - 1)
        f.Close()
        Dim finestra As New Votar(desti, elements)
        finestra.ShowDialog()
        log.Info("La imatge s'ha desat correctament a: " & desti)
    End If
End If
```

Figura 6.24: Restaurar remot.

Per a restaurar un arxiu remot, es recupera l'array de bytes que forma l'arxiu, es crea un FileStream per tal de poder escriure els bytes a l'arxiu, es guarda i s'obre la finestra de valoració.

## Local

```
ElseIf (accio = PyramidAction.ACTION.LOCAL_COPY Or accio = PyramidAction.ACTION.LOCAL_TRANSFORMATION) Then
    If Not IO.File.Exists(elements.Preservation_destination) Then
        MessageBox.Show(My.Application.getMessage("CollectionTabForeignComputer") & elements.Preservation_host & vbCrLf & My.Application.getMessage("CollectionTabComputerName") &
            System.Environment.MachineName & vbCrLf & My.Application.getMessage("CollectionTabNoRecuperation"), "Error")
    Else
        IO.File.Copy(elements.Preservation_destination, desti + "/" + elements.Preservation_Id.ToString + "_" + GetFileName(elements.Preservation_destination), True)
        Dim finestra As New Votar(desti, elements)
        finestra.ShowDialog()
        log.Info("La imatge s'ha desat correctament a: " & desti)
    End If
```

Figura 6.25: Restaurar local.

En la recuperació local es fa com la preservació, es recupera l'arxiu d'allà on sigui i es fa un *File.Copy* de l'arxiu a la carpeta destí.

## Amic

```
Else
    Dim sendOrReceive As New PyramidSenderReceiver()
    Dim restore As Integer = sendOrReceive.Restore(elements.Preservation_Id, elements.Preservation_file.file_id, elements.Preservation_file.file_Collection.collection_id,
        elements.Preservation_file.file_Collection.collection_user_id)
    If restore > 0 Then
        MessageBox.Show(My.Application.getMessage("CollectionTabRestoreRequest"))
    End If
End If
```

Figura 6.26: Restaurar amic.

En aquest cas, s'envia un avís al pyramid de l'amic i, quan aquest es connecti, retornarà automàticament l'arxiu que es demana.

## 6.4 Mòdul multiidioma

Aquest mòdul consisteix en la implementació d'un patró "Visitor" que controla una col·lecció de fitxers que contenen tots els textos de l'aplicació. Cada text està codificat amb una cadena de text que l'identifica de manera que l'aplicació vagi a buscar el text associat a aquest identificador dins del fitxer d'un idioma determinat. L'elecció del patró *Visitor* no és deliberada, sinó que està pensada per tal que a partir d'ara simplement afegint un nou fitxer amb els textos traduïts sigui immediata l'ampliació de l'aplicació amb un nou llenguatge disponible. D'aquesta manera es pot enriquir i ampliar el sistema sense tenir que tocar l'algorítmica de la plataforma. Només és necessari introduir el nou llenguatge i situant el fitxer de textos al directori on el *Visitor* els va a cercar.

## 6.4.1 Implementació Visitor

Per començar, a la carpeta Resources creem uns arxius xml (un arxiu per a cada idioma) on hi haurà els missatges amb un id.

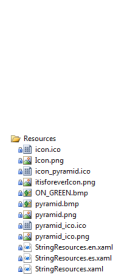


Figura 6.27: Llista de recursos.

```
<system:String x:Key="user">PERFIL</system:String>
<system:String x:Key="collections">ALBUMS</system:String>
<system:String x:Key="options">PREFERÈNCIES</system:String>
<system:String x:Key="friends">AMICS</system:String>
<system:String x:Key="close">Tancar</system:String>
<system:String x:Key="login">Login</system:String>
<system:String x:Key="password">Contrasenya</system:String>

<system:String x:Key="Accept">Acceptar</system:String>
<system:String x:Key="Cancel">Cancel·lar</system:String>
<system:String x:Key="Refuse">Rebutjar</system:String>
<system:String x:Key="Delete">Eliminar</system:String>
<system:String x:Key="Add">Afegir</system:String>
<system:String x:Key="Modify">Modificar</system:String>

<system:String x:Key="AcceptedRefusedFriendRequest">El següent usuari vol ser amic teu:</system:String>
<system:String x:Key="AcceptedRefusedFriendNewFriend">Mou Amic</system:String>

<system:String x:Key="DontRememberPasswordInsertMail">Introduir el Correu-e</system:String>
<system:String x:Key="DontRememberPasswordChanged">Contrasenya canviada! Comprova el teu correu</system:String>
<system:String x:Key="DontRememberChangePassword">Canviar Contrasenya</system:String>
```

Figura 6.28: Missatges en català.

A continuació es crea un mètode a la classe Application (s’hi podrà accedir a través de *My.Application*) que es digui *getMessage* i li passem per paràmetre la cadena de caràcters que identifica el missatge que volem.

I, per saber quin arxiu hem d’agafar, es defineix el llenguatge a les propietats i, depenent del llenguatge seleccionat, s’agafarà un arxiu o altre.

```
Public Sub SetLanguageDirectory()
    Dim dict As New ResourceDictionary
    System.Threading.Thread.CurrentThread.CurrentCulture = New System.Globalization.CultureInfo(My.Settings.idioma)

    Select Case (System.Threading.Thread.CurrentThread.CurrentCulture.Parent.ToString)

        Case "en"
            Me.Resources.MergedDictionaries(4).Source = New Uri("../Resources/StringResources.en.xml", UriKind.Relative)
            dict.Source = New Uri("../Resources/StringResources.en.xml", UriKind.Relative)
            'TODO: Warning!!! break;

        Case "es"
            Me.Resources.MergedDictionaries(4).Source = New Uri("../Resources/StringResources.es.xml", UriKind.Relative)
            dict.Source = New Uri("../Resources/StringResources.es.xml", UriKind.Relative)
            'TODO: Warning!!! break;

        Case "ca"
            Me.Resources.MergedDictionaries(4).Source = New Uri("../Resources/StringResources.xml", UriKind.Relative)
            dict.Source = New Uri("../Resources/StringResources.xml", UriKind.Relative)
            'TODO: Warning!!! break;

        Case Else
            Me.Resources.MergedDictionaries(4).Source = New Uri("../Resources/StringResources.en.xml", UriKind.Relative)
            dict.Source = New Uri("../Resources/StringResources.xml", UriKind.Relative)
            'TODO: Warning!!! break;

    End Select

    Me.Resources.MergedDictionaries.Add(dict)

End Sub

Public Function getMessage(ByVal messageKey As String) As String
    Dim return As String = ""
    If Me.Resources.Contains(messageKey) Then
        return = Me.FindResource(messageKey).ToString
    End If
    Return return
End Function
```

Figura 6.29: Escollir missatge.

## 6.4.2 Com escollir idioma

L'usuari ha d'anar a l'apartat de preferències de l'aplicació i, a baix de tot, hi ha un desplegable per a escollir l'idioma.

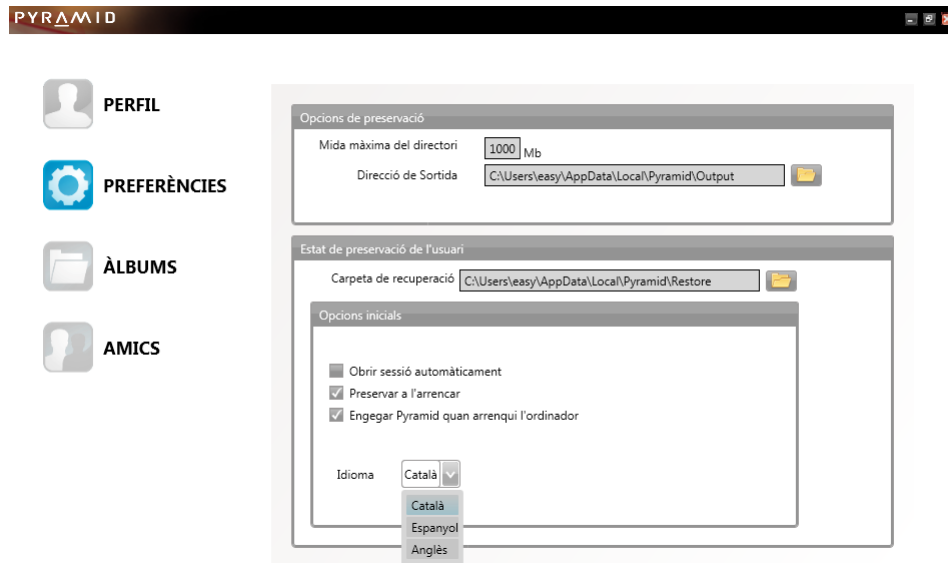


Figura 6.30: Escollir idioma.

Un cop l'usuari ha seleccionat l'idioma, salta un esdeveniment amb el següent codi:

```
Private Sub cboxLanguage_currentItemChanged(ByVal sender As Object, ByVal e As SelectionChangedEventArgs) Handles cboxLanguage.SelectionChanged
    If Not first Then
        If cboxLanguage.SelectedIndex = 0 Then
            System.Threading.Thread.CurrentThread.CurrentCulture = New System.Globalization.CultureInfo("ca-ES")
            My.Settings.idioma = "ca-ES"
            My.Settings.Save()
        ElseIf cboxLanguage.SelectedIndex = 1 Then
            System.Threading.Thread.CurrentThread.CurrentCulture = New System.Globalization.CultureInfo("es-ES")
            My.Settings.idioma = "es-ES"
            My.Settings.Save()
        Else
            System.Threading.Thread.CurrentThread.CurrentCulture = New System.Globalization.CultureInfo("en-GB")
            My.Settings.idioma = "en-GB"
            My.Settings.Save()
        End If
    End If
    My.Application.SetLanguageDirectory()
End Sub
```

Figura 6.31: Codi d'escollir idioma.

Aquest codi captura l'esdeveniment de que hi ha un objecte de la llista d'idiomes seleccionat, recupera l'idioma i crida al mètode ensenyat al subapartat anterior que assigna el recurs corresponent a l'idioma.

## 6.5 Log

Com a tota aplicació, es necessita un registre de què passa al sistema i quins errors dona per a poder solventar-los amb rapidesa. Com que aquesta aplicació treballa en diferents fils d'execució, necessitem implementar un patró *Singleton* (instància única) per tal de que tota la informació es guardi només en un únic arxiu.

Així doncs, la classe Log tindrà la següent forma:

```
#Region "Private Attributes"
Private log_ As ILog
#End Region

Private Shared instance As Log
Public Shared ReadOnly Property Instance As Log
    Get
        If Log.instance Is Nothing Then
            instance = New Log
        End If
        Return instance
    End Get
End Property
Private Sub New()
    XmlConfigurator.Configure(New System.IO.FileInfo(AppDomain.CurrentDomain.BaseDirectory + "log4net.xml"))
    log_ = log4net.LogManager.GetLogger("Pyramid")
End Sub
#End Region
#Region "Metodes"
Public Function Err(ByVal Missatge As String)
End Function
Public Function Err(ByVal Missatge As String, ByVal ex As Exception)
End Function
Public Function FatalErr(ByVal Missatge As String)
End Function
Public Function FatalErr(ByVal Missatge As String, ByVal ex As Exception)
End Function
Public Function Debug(ByVal Missatge As String)
End Function
Public Function Debug(ByVal Missatge As String, ByVal ex As Exception)
End Function
Public Function Warn(ByVal Missatge As String)
End Function
Public Function Warn(ByVal Missatge As String, ByVal ex As Exception)
End Function
Public Function Info(ByVal Missatge As String)
End Function
Public Function Info(ByVal Missatge As String, ByVal ex As Exception)
End Function
#End Region
End Class
```

Figura 6.32: Classe Log.

Hi haurà un atribut privat que serà l'objecte log, cada cop que es vulgui instanciar, es comprovarà que no existeixi ja, si és així, s'instancia i es retorna, en cas contrari, només es retorna.

Així, cada cop que es vulgui tenir accés al registre, des del mètode pertinent s'haurà de fer "Dim log As Log = log.Instance" i ens retornarà el mateix arxiu de log a tot arreu.

Com es pot comprovar a la imatge, tenim diferents mètodes per escriure depenent del grau d'importància, tenim mètodes d'informació, d'error, de *warning*, etc... Que es veuran reflectits en el mateix registre.

## 6.6 Proves

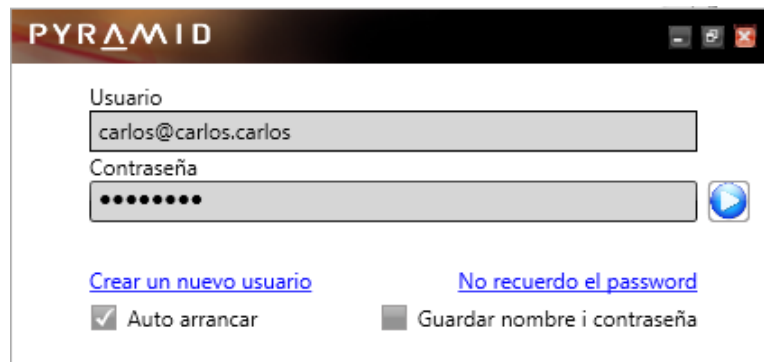
En aquest apartat es faran algunes proves per veure que l'aplicació funciona correctament i que compleix els requeriments.

### 6.6.1 Autenticar l'usuari

Tenim una pantalla de Login que ens deixa entrar al sistema amb usuari i contrasenya i registrar-nos.

#### 6.6.1.1 Login erroni

A continuació es veu què passa si s'introdueix un login erroni.



The screenshot shows a window titled "PYRAMID" with a login form. The "Usuario" field contains "carlos@carlos.carlos" and the "Contraseña" field is filled with dots. Below the fields are two links: "Crear un nuevo usuario" and "No recuerdo el password". At the bottom, there are two checkboxes: "Auto arrancar" (checked) and "Guardar nombre i contraseña" (unchecked). A blue play button is visible to the right of the password field.

Figura 6.33: Login erroni.

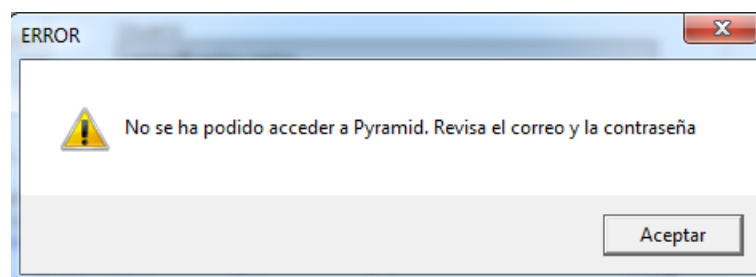


Figura 6.34: Login error.



### 6.6.1.2 Login correcte

En canvi, si el login és correcte, entra a la pantalla principal de l'aplicació.

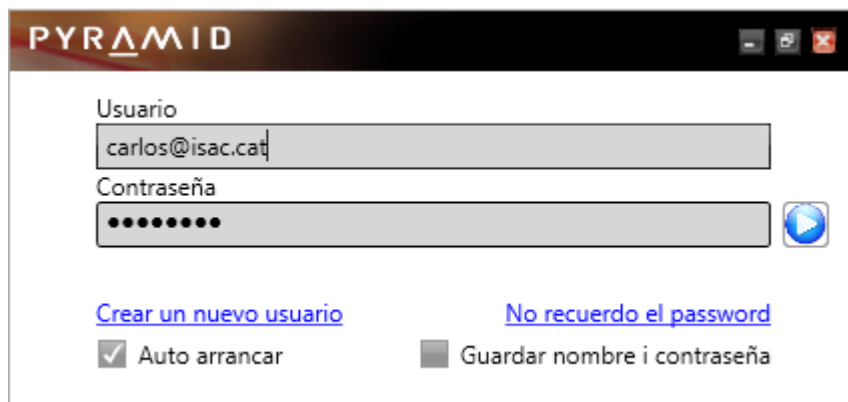


Figura 6.35: Login correcte.

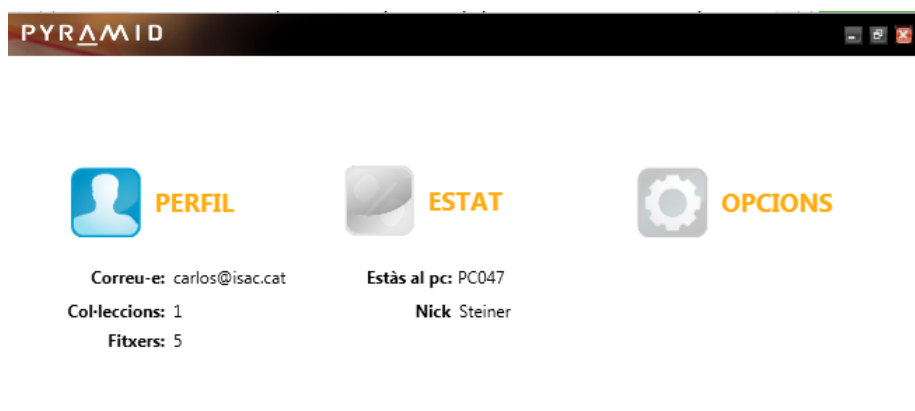


Figura 6.36: Pantalla principal.

## 6.6.2 Administrar àlbums

Tenim un sistema que ens permet crear, modificar, eliminar àlbums. Ens permet veure els elements preservats, els que no, les preservacions que s'han fet i la vista en miniatura de l'original.

### 6.6.2.1 Crear àlbum

En la creació d'un àlbum ens demana la ubicació del directori, a continuació ens demana que guardem l'àlbum. Podem modificar el nom d'aquest quan volguem i clicar a guardar.

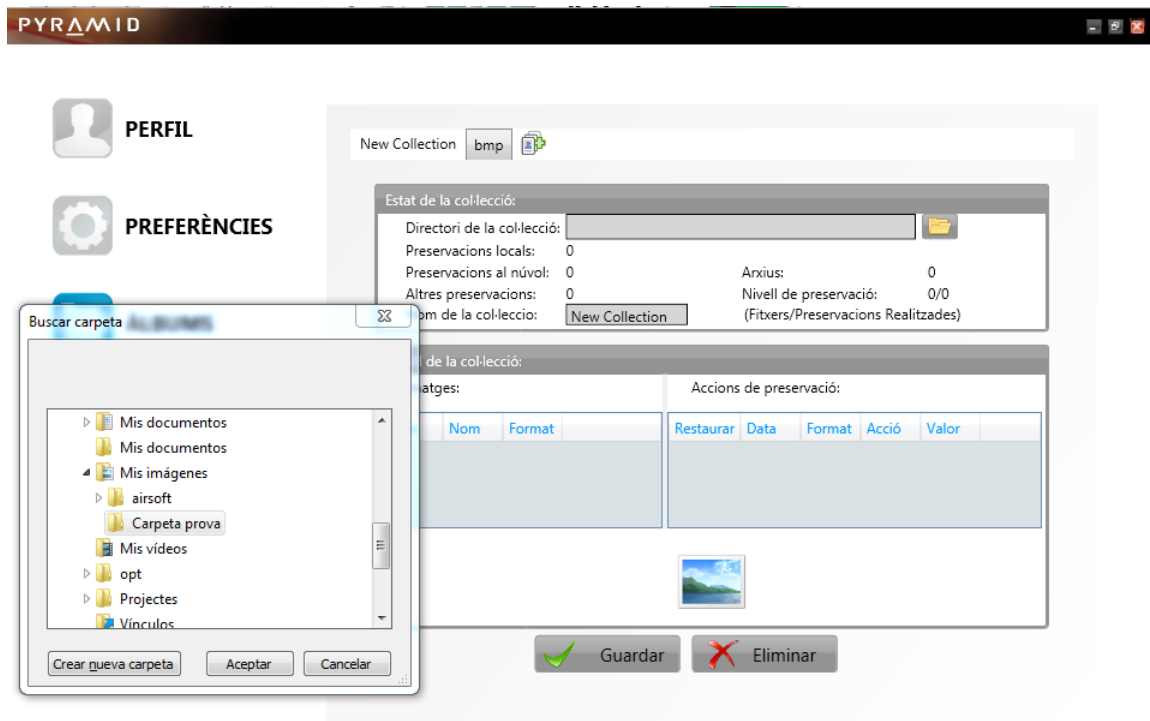


Figura 6.37: Nou Album.

### 6.6.2.2 Modificar àlbum

Qualsevol modificació no es tindrà en compte fins a clicar al botó guardar. Realment només es pot modificar desde l'aplicació el nom de l'aplicació, si es volen afegir o treure elements de l'àlbum s'ha de fer dirèctament des del directori origen.

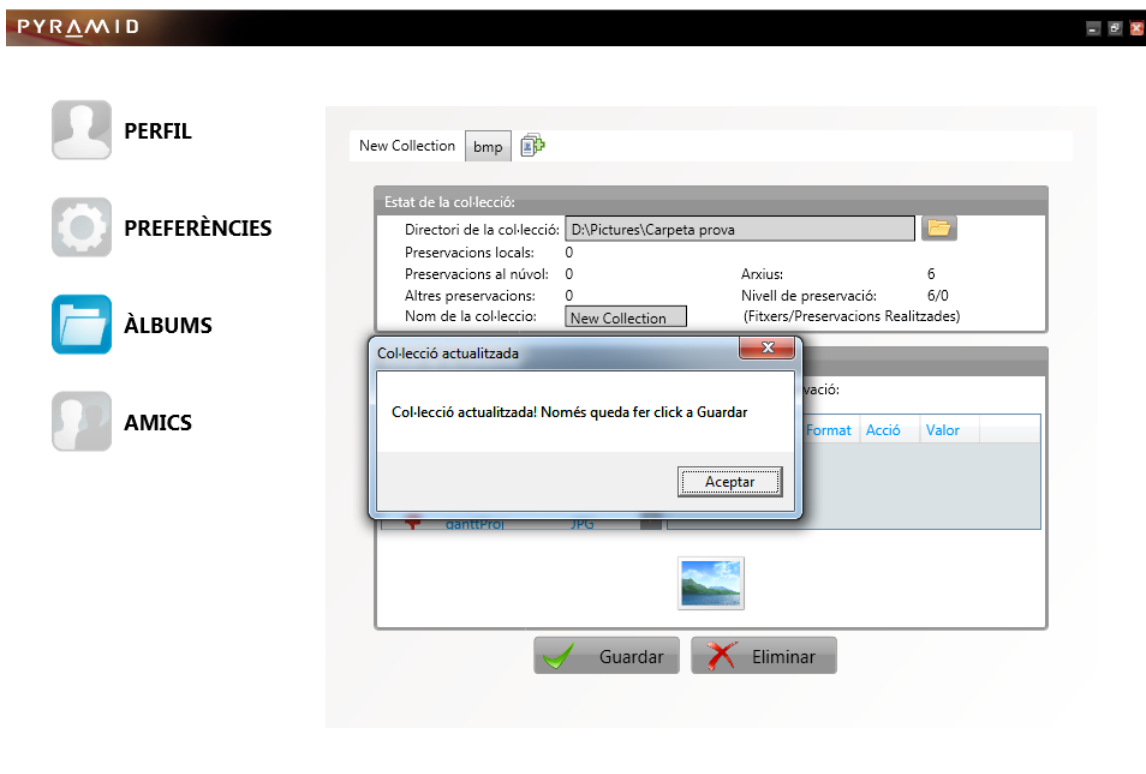


Figura 6.38: Guardar Àlbum.

### 6.6.2.3 Vista en miniatura

En fer clic en qualsevol element de la llista, podem veure una vista en miniatura.

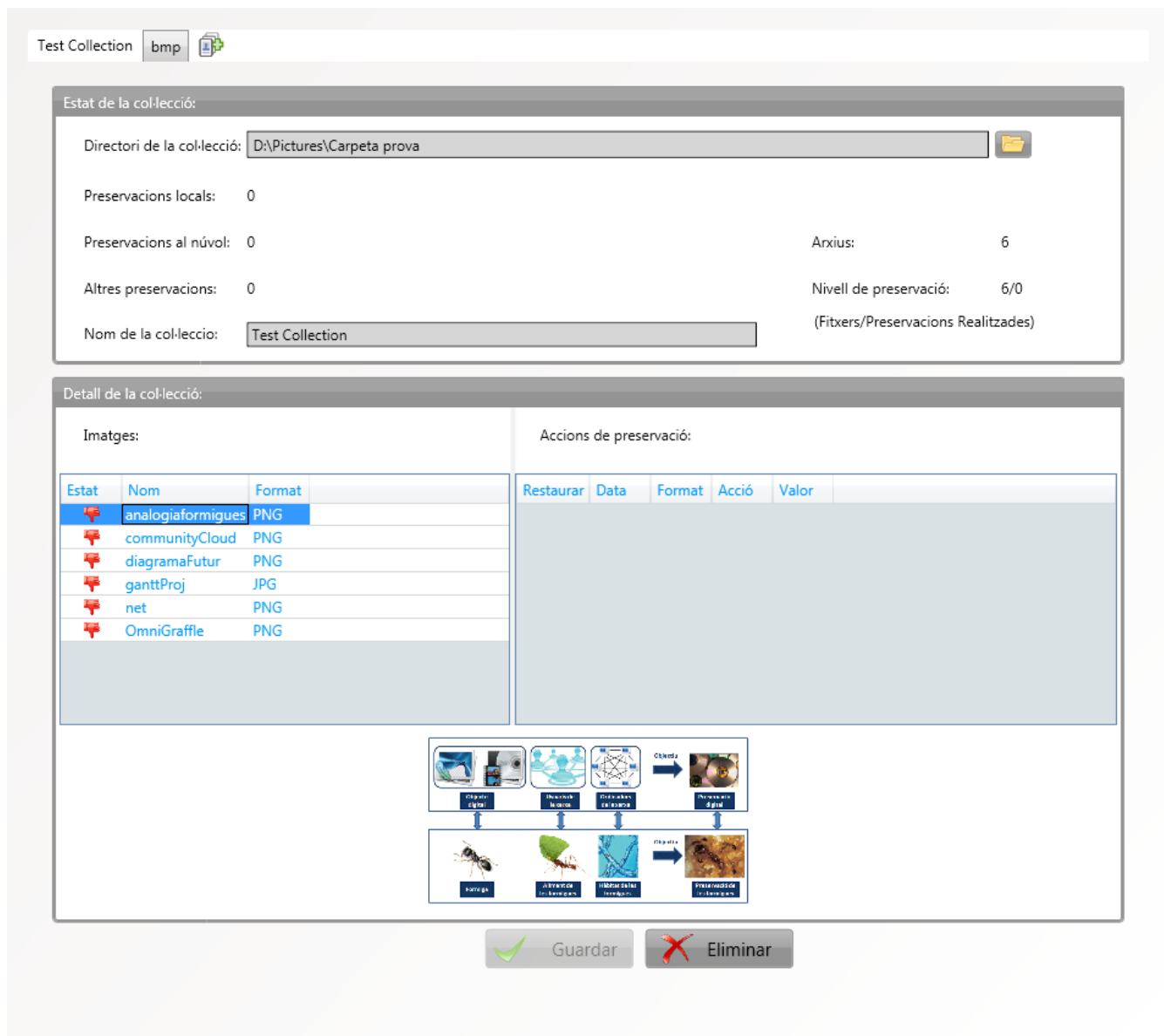


Figura 6.39: Vista en Miniatura.

#### 6.6.2.4 Eliminar álbum

A la primera imatge hem volgut eliminar el primer àlbum, a la segona podem comprovar que ja no hi és.

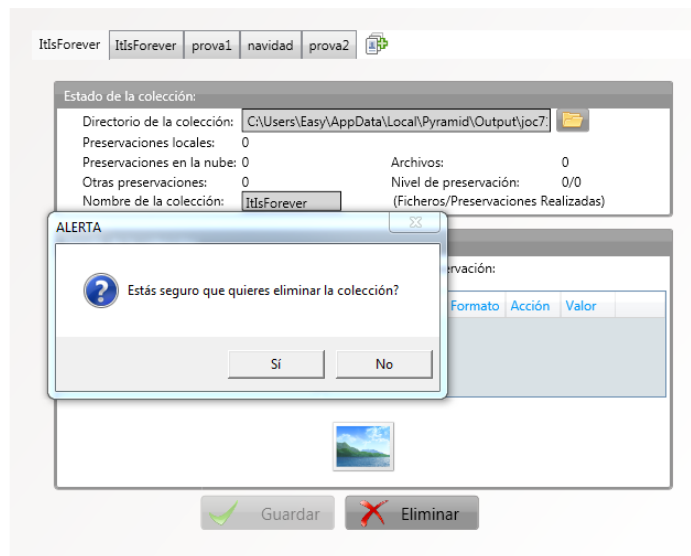


Figura 6.40: Eliminar àlbum.

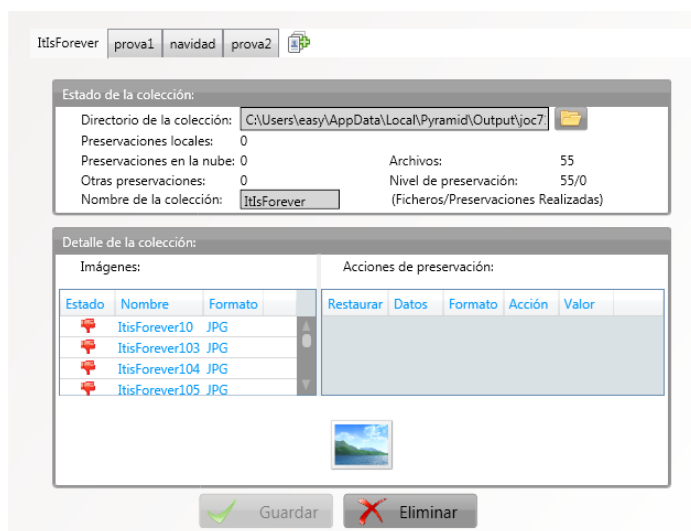


Figura 6.41: Àlbum Eliminat.

### 6.6.2.5 Confirmació visual de canvi

A la següent imatge es veu com ha aparegut un element nou, que hem afegit prèviament al directori de l'àlbum.

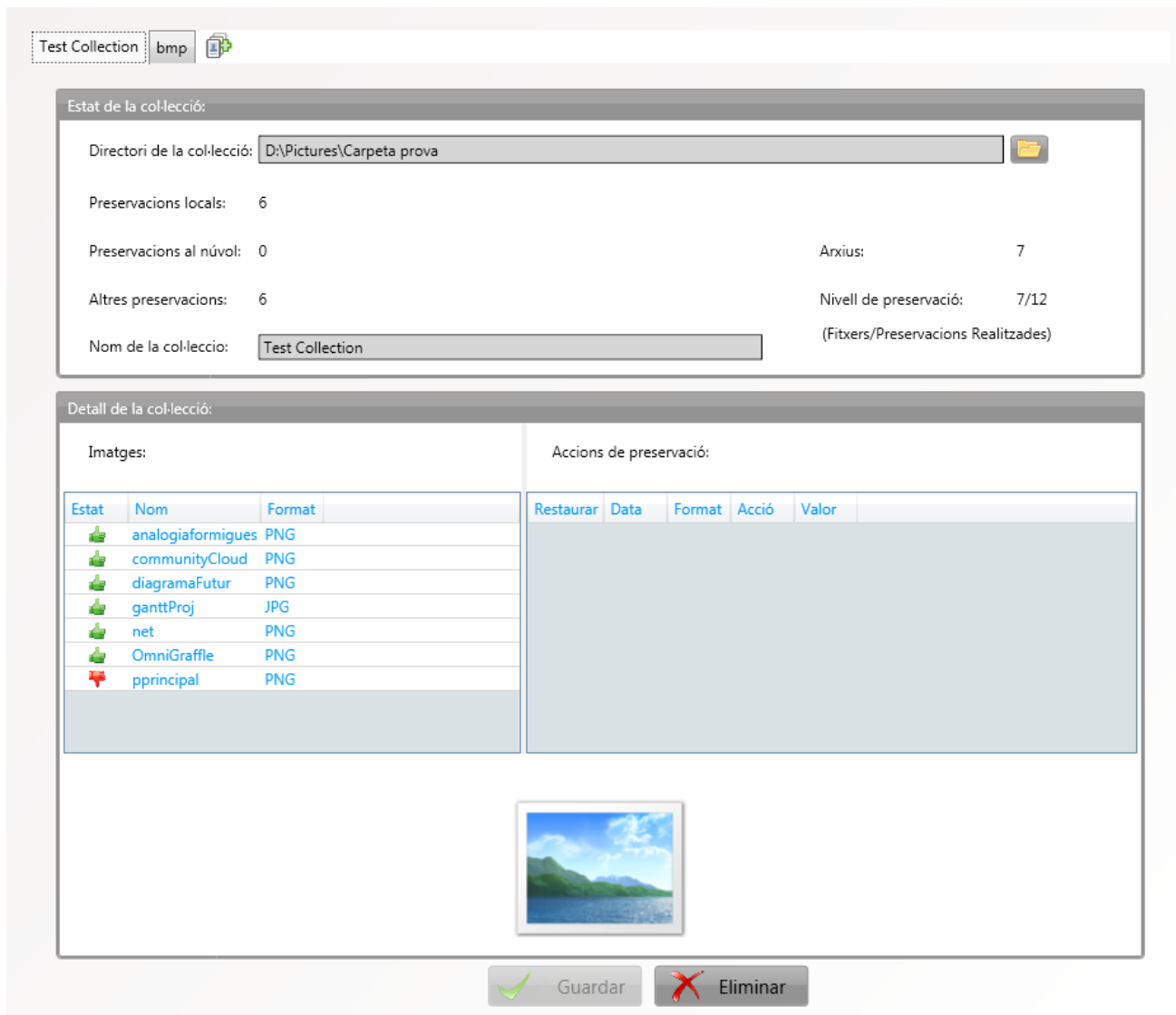


Figura 6.42: Nou Element.

### 6.6.3 Preservació i restauració

A continuació es deixa constància de que realment es preserva i es poden recuperar/restaurar aquestes preservacions.

A la figura 6.43 es veu que s'han realitzat còpies i transformacions tan en local com en remot.

Detall de la col·lecció:

Imatges:			Accions de preservació:				
Estat	Nom	Format	Restaurar	Data	Format	Acció	Valor
👍	analogiaformigues	PNG	📁	21/05/2013 11:44	jpg	Còpia remota	0
👍	communityCloud	PNG	📁	21/05/2013 11:25	jpg	Còpia local	0
👍	diagramaFutur	PNG	📁	21/05/2013 11:41	tiff	Transformació local	0
👍	ganttProj	JPG	📁	21/05/2013 11:38	png	Transformació remota	0
👍	net	PNG	📁	21/05/2013 11:39	jpeg	Transformació local	0
👍	OmniGraffle	PNG	📁	21/05/2013 11:45	bmp	Transformació remota	0
👍	pprincipal	PNG					




Figura 6.43: Diferents preservacions.

### 6.6.3.1 Preservacions

A la pàgina anterior (figura 6.43) es veu que les primeres preservacions són una còpia remota i una de local, a les següents figures es deixa constància de que és així.

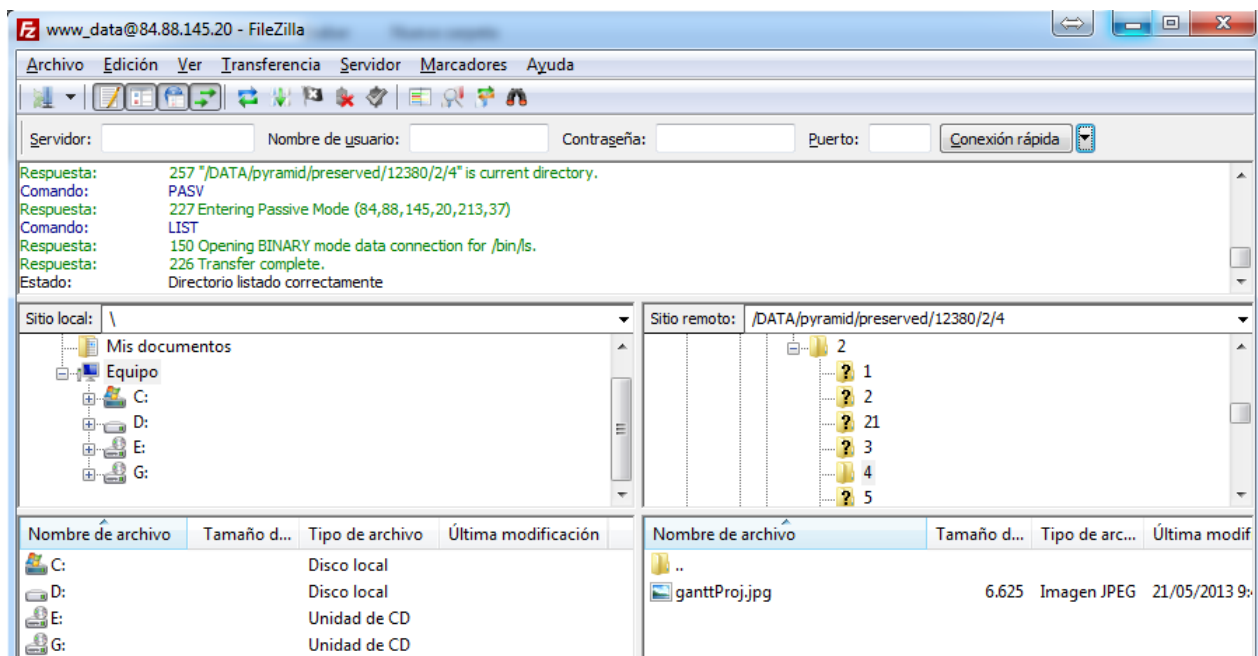


Figura 6.44: Preservació remota

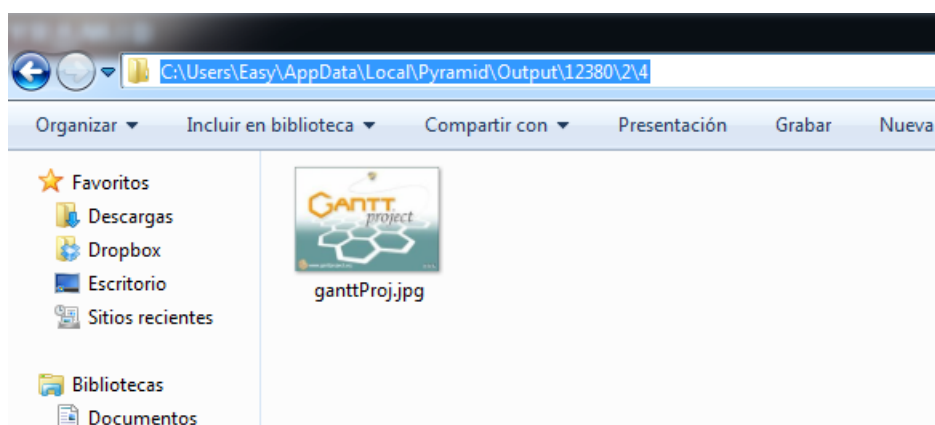


Figura 6.45: Preservació local



### 6.6.3.2 Restaurar

Per a recuperar una imatge que hem perdut, hem de fer clic dret sobre l'arxiu que volem recuperar i escollir restaurar arxiu (com es veu a la figura 6.46), finalment ens apareix una finestra (figura 6.47) amb l'enllaç cap a l'arxiu restaurat i ens demana que puntuem la restauració per tal d'actualitzar estadístiques i que la comunitat sàpiga quin preservador és millor.



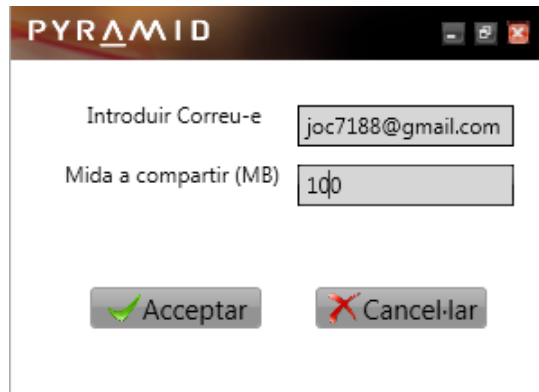
Figura 6.46: Petició de restauració.



Figura 6.47: Restaurat.

## 6.6.4 Amics i formats de moda

Desde la pantalla d'amics, si cliquem al botó afegir ens apareix la següent finestra on hem d'afegir correu-e i tamany de disc que li donem a l'amic.



The screenshot shows a window titled "PYRAMID" with a dark header. The main content area is white and contains two input fields. The first field is labeled "Introduir Correu-e" and contains the text "joc7188@gmail.com". The second field is labeled "Mida a compartir (MB)" and contains the text "100". Below the input fields are two buttons: "Acceptar" with a green checkmark icon and "Cancel·lar" with a red X icon.

Figura 6.48: Afegir amic.

A l'amic, si te compte pyramid, en obrir sessió li apareix la següent finestra, en cas contrari, li arriba un correu-e.



The screenshot shows a window titled "PYRAMID" with a dark header. The main content area is white and contains the text "El següent usuari vol ser amic teu:" followed by the email address "carlos@isac.cat". Below this text are two buttons: "Acceptar" with a green checkmark icon and "Rebutjar" with a red X icon.

Figura 6.49: Petició amic.

### 6.6.4.1 Formats de moda

Un cop el segon usuari ha acceptat la petició, li apareix la següent imatge on hi ha la llista d'amics i, clicant en cada un d'ells, pots veure quins són els formats que més fan servir.

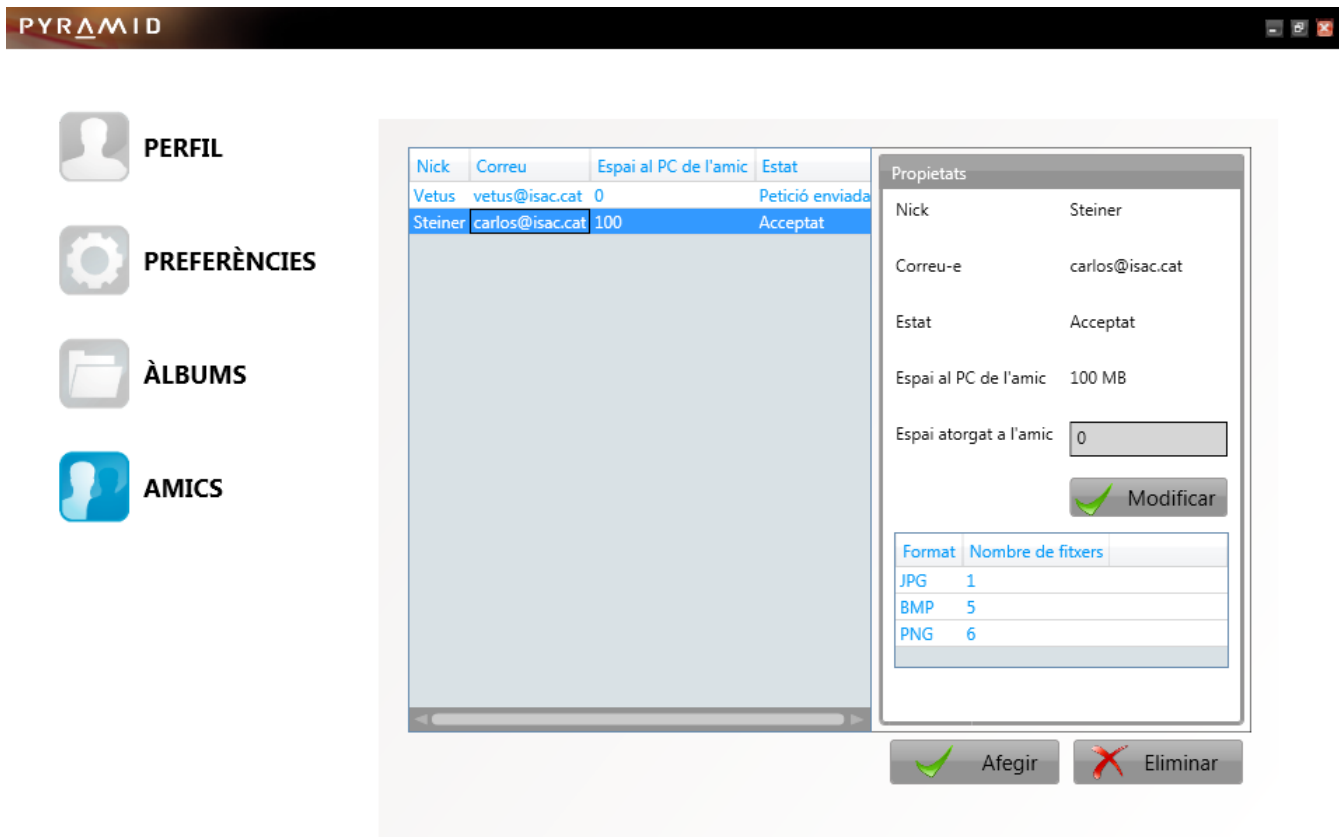


Figura 6.50: Amics i formats

## 6.6.5 Multiidioma

Un altre dels requeriments que s'havien establert, era que l'aplicació havia de ser multiidioma.

### 6.6.5.1 Català

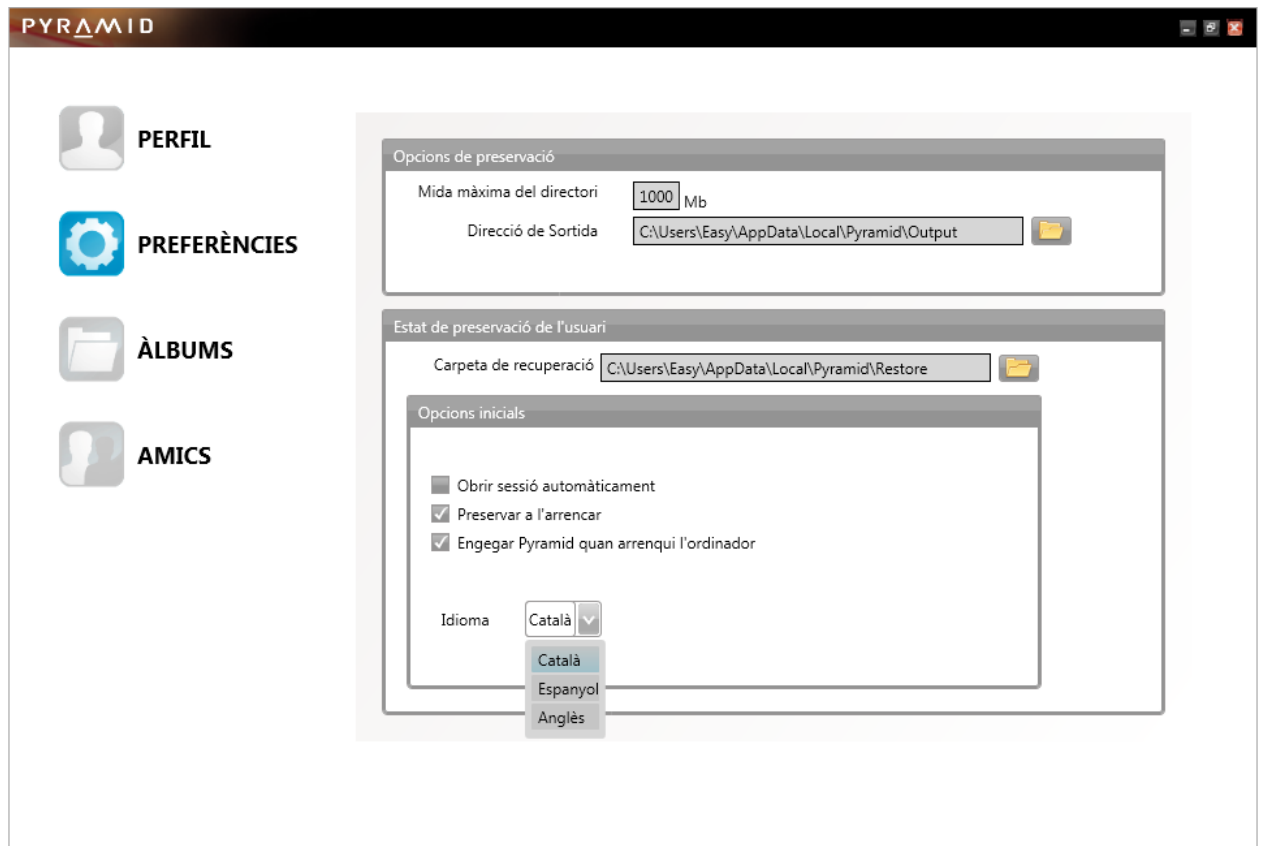


Figura 6.51: Aplicació en català

### 6.6.5.2 Anglés

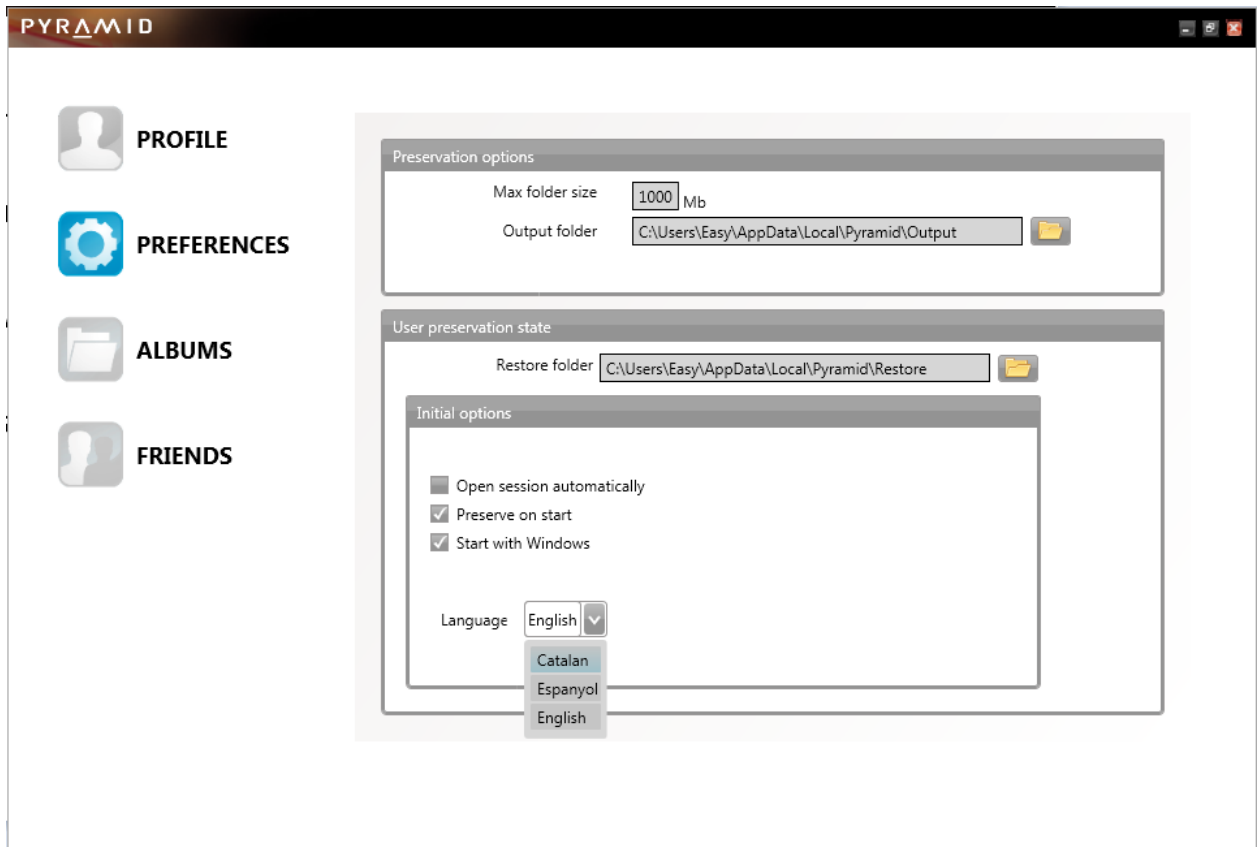


Figura 6.52: Aplicació en anglés

### 6.6.6 Ampliació dinàmica de preservadors

Com es veia al diagrama de classes, tenim preservador còpia, preservador que transforma de BMP a altres formats i preservador que transforma de MTIF a altres formats.

S'ha creat un preservador extern de JPG a altres formats per a provar que l'ampliació dinàmica funciona.

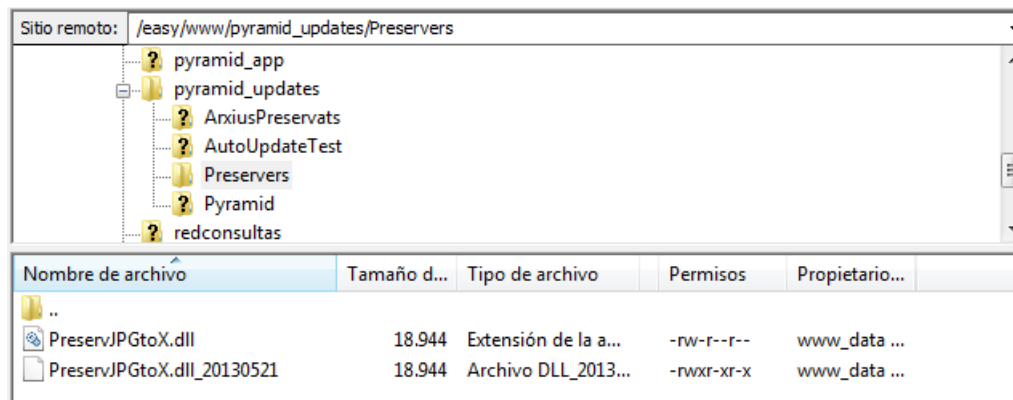


Figura 6.53: DLL al servidor

Estado	Nombre	Formato	Restaurar	Datos	Formato	Acción
	communityCloud	PNG		21/05/2013 11:25	jpg	Copia local
	diagramaFutur	PNG		21/05/2013 11:47	tiff	Transformación local
	ganttProj	JPG		21/05/2013 11:38	png	Transformación remota
	net	PNG		21/05/2013 11:46	jpeg	Transformación local
	OmniGraffle	PNG		21/05/2013 11:48	bmp	Transformación remota
	pprincipal	PNG				

Figura 6.54: Imatge jpg transformada a altres formats

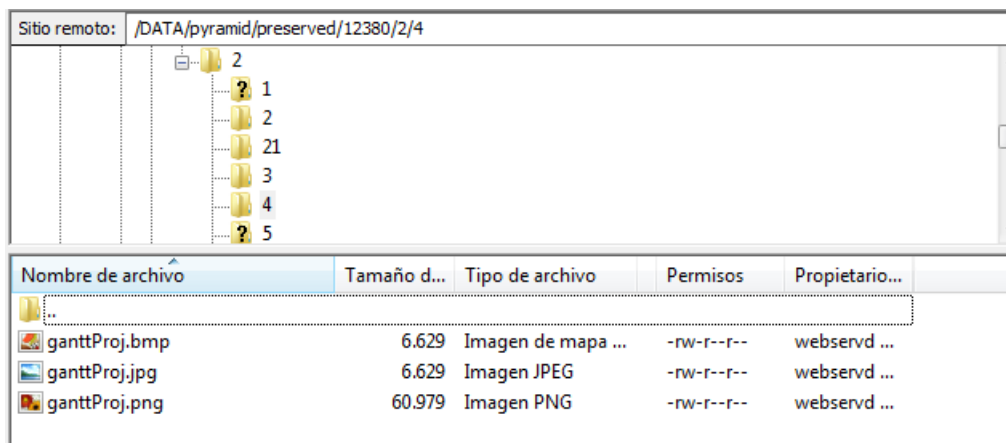


Figura 6.55: Transformacions al servidor

### 6.6.7 Execució en *Background*

Un altre dels requeriments era que l'aplicació es pogués executar en segon pla i que l'usuari no notés que l'aplicació està engegada.



Figura 6.56: Aplicació en segon pla

Pyramid.vshost.exe *32	Easy	13	56.564 KB	vshost32.exe
RtkNGUI64.exe	Easy	00	5.512 KB	Realtek HD Audio Ma...
rundll32.exe	Easy	00	3.636 KB	Proceso host de Win...
SearchIndexer.exe	SYSTEM	00	43.704 KB	Indizador de Microsof...
services.exe	SYSTEM	00	3.268 KB	Aplicación de servicio...
Skype.exe *32	Easy	00	99.072 KB	Skype

Figura 6.57: Consum de l'aplicació

# Capítol 7

## Conclusions

En aquest apartat s'avaluaran els objectius i requeriments que es van proposar a l'inici del projecte.

També s'explicarà algun dels problemes trobats i què es podria millorar en un futur per a que el Pyramid fos una millor eina de preservació.

### 7.1 Avaluació dels objectius

Recordem que els objectius finals eren:

- **Implementació d'una aplicació per a la PD d'us domèstic:** Aquest objectiu trobo que ha estat completat ja que tenim l'aplicació penjada a la web i està en funcionament a diversos ordinadors.
- **Aprofitar el motor d'aquesta aplicació per a la implementació d'altres serveis de PD:** A dia d'avui s'ha fet servir com a motor de preservació per un servei web anomenat **It Is Forever**, aquest servei s'explica als annexos.
- **Permetre, gràcies a l'estudi i l'elaboració d'aquest projecte, continuar investigant sobre la PD i/o tasques relacionades amb aquesta:** Tal com està la aplicació, dona pas a investigar sobre millores d'aquesta i ja hi ha en procés estudis sobre la millora de la PD amb les tècniques esmentades al segon capítol.

Els objectius principals es van haver de resoldre abans de continuar amb els objectius finals.



## 7.2 Avaluació dels requeriments

A continuació farem una valoració de l'assoliment dels requeriments tan funcionals com no funcionals.

### 7.2.1 Requeriments funcionals

1. **Autenticar l'usuari al sistema (Login):** Tenim un sistema de login amb correu-e i contrassenya que apareix abans de començar a fer servir l'aplicació.
2. **Administrar col·leccions (àlbums):** Hi ha un apartat àlbums que et permet afegir, modificar i/o eliminar àlbums.
3. **Backups locals o remots de les col·leccions:** Realment es fan còpies tan locals com remotes, però no de les col·leccions, si no dels arxius que componen aquesta col·lecció.
4. **Transformacions locals o remotes de les col·leccions:** Com al punt anterior, es fan transformacions tan locals com remotes.
5. **Restauració fidedigne de les preservacions:** Es poden restaurar els arxius de forma fidedigne, l'usuari puntuarà la qualitat de la restauració.
6. **Previsualització de l'arxiu:** En clicar sobre qualsevol arxiu d'una col·lecció, t'apareix una vista en miniatura d'aquest.
7. **Aplicar accions de preservació locals o remotes de forma intel·ligent:** Aquest és potser el requeriment que més s'hauria de valorar, fa servir de forma intel·ligent la preservació entre amics, entre les locals o remotes no discrimina gaire bé quins arxius preservar.
8. **Obtenir coneixement de la comunitat sobre els formats de moda:** La base de dades té uns camps amb les puntuacions que s'han donat als formats de preservació i la quantitat d'arxius de cada format que hi ha al sistema.
9. **Llista d'amics:** Es poden afegir i eliminar amics a través de correu electrònic.

### 7.2.2 Requeriments no funcionals

1. **Interfície Multiidioma:** Gràcies al patró visitor s'ha pogut afegir fàcilment un sistema de missatges multiidioma.

2. **Actualitzacions automàtiques:** En engegar l'aplicació es compara la versió d'aquesta amb la que hi ha a BdD, si és més petita, es descarrega la nova versió i s'instala automàticament.
3. **Ampliació dinàmica dels preservadors:** Mentre el pyramid està engegat, cada x temps es comprova si hi ha algun preservador nou al servidor, el descarrega i, gràcies al sistema creador implementat, en descarregar-se queda totalment funcional.
4. **Interfície senzilla, clara i d'estil 2.0. Fent ènfasis en la usabilitat:** La vista de l'explicació és realment senzilla i en pocs clicks pots fer qualsevol cosa.
5. **Possibilitat d'executar-se en background amb un *tray icon* a la barra d'estat:** Un cop tanques la finestra, el programa s'executa en segon pla, deixant una icona a la barra d'estat per si es vol tornar a obrir, així continua preservant sense que l'usuari se n'adoni.
6. **L'aplicació ha d'ésser lleugera, per tan, l'usuari no ha de notar cap relentiment del sistema quan aquesta estigui en funcionament:** L'aplicació en *Background* no ocupa gaires recursos, l'usuari no nota una baixada de rendiment del sistema.
7. **Base de dades compacta i portable:** La base de dades s'ha fet en MySQL, cosa que facilita el fet de fer backups i portabilitzar-la a altres servidors.
8. **Informar de forma visual quan s'han realitzat operacions o s'han detectat nous arxius a les col·leccions:** A la pantalla d'àlbums es pot veure quans fitxers es tenen i quantes preservacions se n'han fet.

## 7.3 Problemes trobats

El problema principal que ens vam trobar a l'hora de realitzar el projecte va ser en el mòdul de preservació entre amics.

El *Firewall* (tallafocs) va ser el nostre principal problema per aconseguir enviar els arxius preservats als nostres amics.

Per solventar-ho, es va intentar afegir un petit mòdul que enviés les dades per UDP però també ens donava problemes el fet que s'havia d'estar escoltant constantment i això, a més de gastar més recursos, deixava un port inutilitzat. Per tan, en comptes d'enviar els arxius directament entre dos pc amb *Pyramid*, s'ha de fer un pas intermig amb el servidor, cosa que relentitza el procés de preservació i restauració d'aquest mode.

Un altre problema ha sigut com introduir els agents en el projecte, en aquesta versió pre-alpha hem vist que era més eficient fer servir unes funcions que interactuen amb la Bdd en comptes de desenvolupar els agents pròpiament dits, que s'implementarien en futures versions quan l'aplicació pyramid estés més extesa a les llars dels usuaris i tinguèssim una comunitat per a treure profit d'aquests agents (a l'annex B s'expliquen els principals problemes de treballar amb agents).

## 7.4 Treball futur

Per a un futur hi ha diverses coses a millorar, entre elles:

- **Crossplatform:** Adaptar l'aplicació per a que pugui córrer en qualsevol sistema operatiu, ara està feta per windows, per córrer en qualsevol altre sistema, s'hauria de fer us d'un entorn virtual.
- **Facebook i/o Twitter:** Que la llista d'amics es pugui complimentar a través dels teus amics de facebook o twitter.
- **Aplicació mòbil:** Allà on acostumem a fer més fotos en la vida quotidiana i on més les perdem és al mòbil, per tan, aprofitar el motor per a fer un *Pyramid* mòbil seria una bona estratègia de futur.
- **Millorar la preservació entre amics:** Aconseguir eliminar el pas intermig de la preservació entre amics, és a dir, una comunicació directa entre Pyramid i Pyramid sense el servidor de per mig, com veiem a la següent figura.

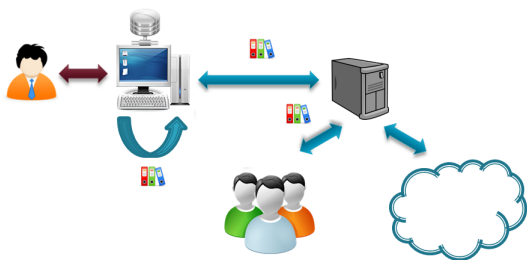


Figura 7.1: Diagrama actualment

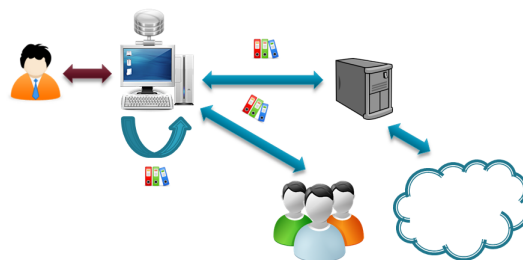


Figura 7.2: Diagrama en el futur.

- **Implementar agents:** La implementació d'agents és una tasca pendent, per tal d'agilitzar i fer més eficient el procés de preservació dins del *Community Cloud*.

- **Afegir preservadors:** Afegir més preservadors per tal d'augmentar l'interés en l'aplicació, aquesta millora no te gaire dificultat gràcies al mòdul d'ampliació de preservadors que simplement creant una nova DLL es poden anar afegint dinàmicament.

# Capítol 8

## Manual d'usuari

A continuació s'il·lustrarà com descarregar-se i com fer servir l'aplicació Pyramid.

### 8.1 Descàrrega i instal·lació

Per a obtenir l'aplicació Pyramid hem d'anar a [www.pyramid.cat](http://www.pyramid.cat) i clicar sobre la fletxa blanca amb fons verd que hi ha al costat de la paraula Pyramid a la part inferior de la pantalla.

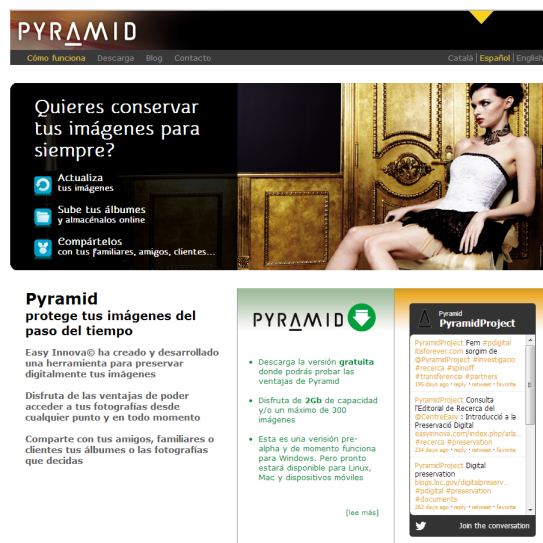


Figura 8.1: Web del pyramid



Figura 8.2: Instal·lador descarregat.

Un cop descarregat l'arxiu *Setup.msi* s'ha de fer doble clic i apareixerà l'assistent d'instal·lació. Només caldrà anar fent clic a "Siguiente", escollint ubicació d'instal·lació i si instal·lar-ho en un usuari o tots els que facin servir el pc fins que surti el missatge de instal·lació correcte.

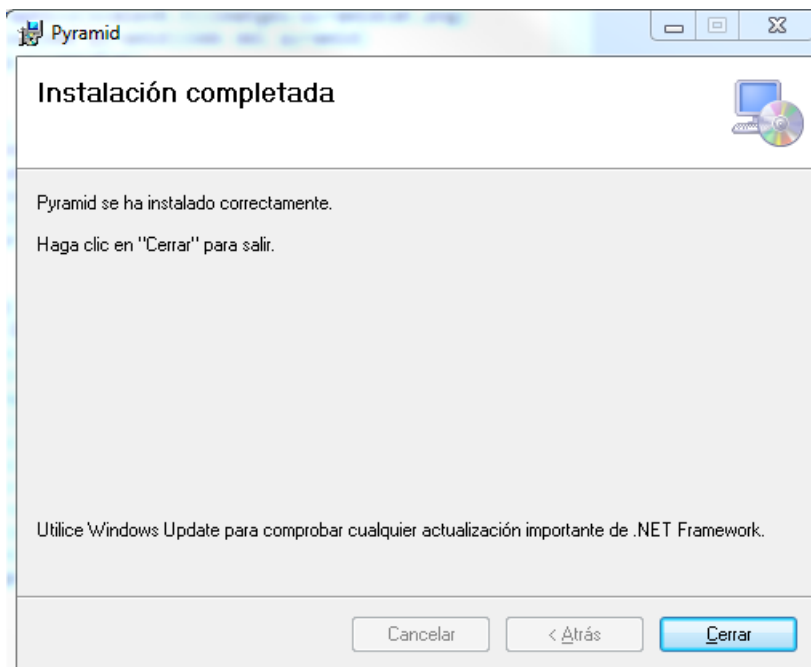


Figura 8.3: Instal·lació correcte.

## 8.2 Utilització

Ja tenim l'aplicació instal·lada, ara només cal obrir-la amb l'accés directe que ha aparegut a l'escriptori i apareixeran les següents pantelles.

### 8.2.1 Pantalla *Login*

A la pantalla de *login* ens apareixen les opcions de registrar-nos, fer *login* pròpiament o recuperar la contrassenya.

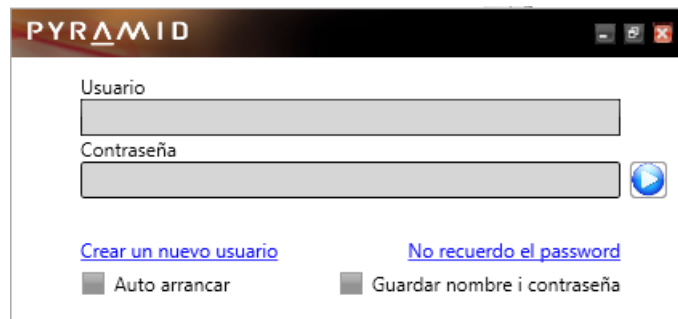


Figura 8.4: Pantalla Login.

En el cas de que ja tinguem un compte i no recordem la contrassenya, en clicar en "No recuerdo el password" se'ns demanarà el correu electrònic i, un cop acceptat, s'enviarà la nova contrassenya via correu electrònic.

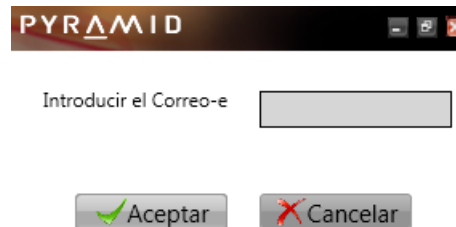



Figura 8.5: Pantalla nou password.

### 8.2.1.1 Registrar

Per a registrar-nos, hem de clicar a crear nuevo usuario i ens apareixerà una nova pantalla per introduir les nostres dades.



The image shows a registration window titled "PYRAMID". It contains three input fields: "Correo-e", "Password", and "Repetir contraseña". Below the fields is a note: "(Contraseña: Máximo 8 caracteres)". At the bottom, there are two buttons: "Aceptar" (with a green checkmark icon) and "Cancelar" (with a red X icon).

Figura 8.6: Pantalla crear usuari.

Un cop entrades les dades i fent clic en acceptar, se'ns enviarà un correu electrònic per verificar que les dades són correctes, un cop s'entra a l'enllaç, ja som oficialment usuaris de Pyramid.

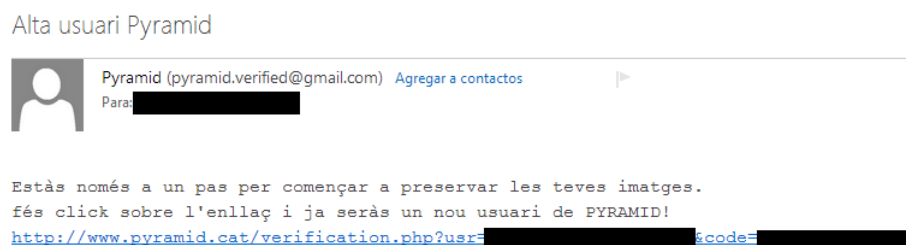


Figura 8.7: Correu de validació.

Ara només queda tornar a la pantalla de *login*, entrar les dades i donar a acceptar.

Un cop acceptat, no apareix directament la pantalla principal, sinó que apareix la icona a la barra d'estat de windows.



## 8.2.2 Pantalla Principal

Per entrar a la pantalla principal s'haurà de fer doble clic a la icona de Pyramid de la barra d'estat.

Llavors ens apareix la pantalla principal amb tres opcions:

- **Perfil:** Ens informa de qui som, les col·leccions i arxiu que tenim, el nostre nick i en quin ordinador estem.
- **Estat:** Conté un botó per aturar i engegar la preservació.
- **Opcions:** Ens obre una nova pantalla amb les diferents opcions.

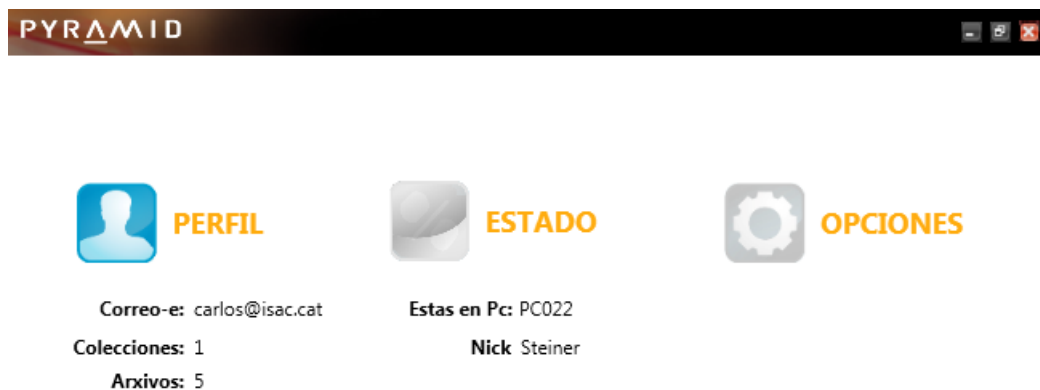


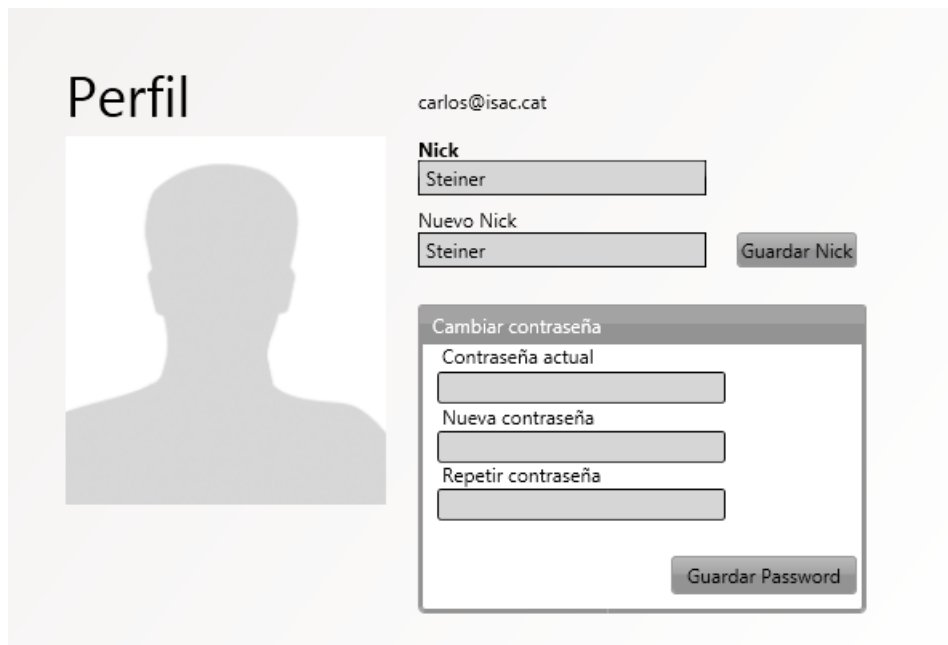
Figura 8.8: Pantalla principal.

## 8.2.3 Pantalla opcions

La pantalla opcions és la mateixa que podem veure a la figura 6.30 que ens permet navegar entre diferents alternatives.

- **Perfil:** Ens permet canviar el nick i la contrassenya.
- **Preferències:** Aquí es poden modificar les configuracions de preservació i restauració, així com canviar l'idioma, etc.
- **Àlbums:** Aquesta opció és la de gestionar les col·leccions.
- **Amics:** Trobem el llistat d'amics i ens deixa afegir i/o eliminar aquests.

### 8.2.3.1 Perfil



The screenshot shows a user profile page titled "Perfil". On the left is a placeholder for a profile picture. To the right, the email address "carlos@isac.cat" is displayed. Below it, there are two input fields for "Nick", both containing the text "Steiner". A "Guardar Nick" button is positioned to the right of the second input field. A modal dialog box titled "Cambiar contraseña" is open, containing three input fields: "Contraseña actual", "Nueva contraseña", and "Repetir contraseña". A "Guardar Password" button is located at the bottom right of the dialog.

Figura 8.9: Pantalla perfil.

En la pantalla de perfil, podem canviar les dades com el Nick, o canviar la contrassenya.

### 8.2.3.2 Preferències

En aquesta pantalla es poden configurar les preferències de l'aplicació:

- **Opcions de preservació:** Assignar el directori on aniran a parar les preservacions locals i limitar l'ocupació de memòria d'aquest.
- **Estat de preservació:** Assignar el directori de restauració.
- **Opcions inicials:** Permet seleccionar opcions com obrir sessió automàticament en engegar el pyramid, que aquest s'engegui preservant o no, que s'engegui el pyramid en encendre l'ordinador i l'idioma de la interfície del pyramid.

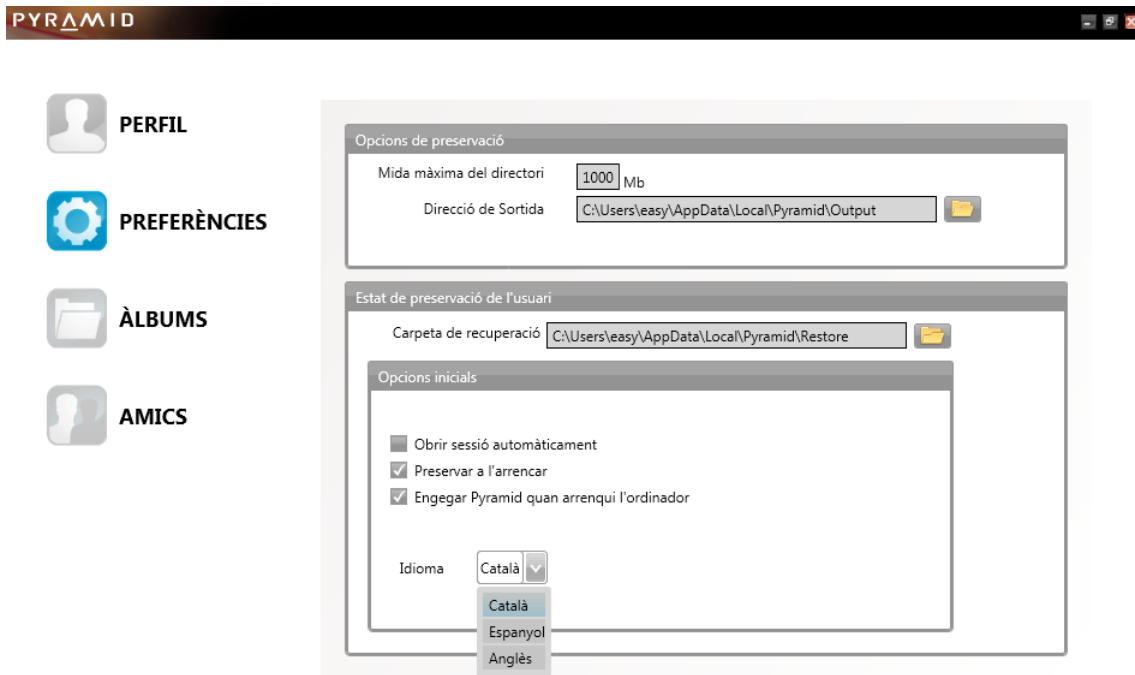


Figura 8.10: Pantalla preferències.

### 8.2.3.3 Àlbums

Aquí es poden gestionar les col·leccions i arxius:

- **Afegir àlbum:** Clicant al símbol ”+” que hi ha a la pestanya, es crea un nou àlbum, llavors has d’escollir la direcció de la col·lecció fent click a la icona de fitxer que obrirà un diàleg per a seleccionar la carpeta.
- **Modificar àlbum:** Es pot modificar el nom de l’àlbum canviant directament el nom a la casella de ”Nom de la col·lecció”. Per a afegir nous arxius a la col·lecció s’haurà de fer afegint-los directament al directori, el pyramid ja detectarà que hi ha nous arxius i els afegirà.
- **Eliminar àlbum:** Un cop seleccionat l’àlbum que es vol eliminar, fas clic en el botó eliminar, confirmes i ja s’haurà eliminat.
- **Restaurar arxiu:** Per a restaurar un arxiu, selecciones l’arxiu a restaurar de la llista de imatges (podràs comprovar que és el que vols amb la vista prèvia de sota), llavors selecciones una de les preservacions que apareix a la llista de la dreta, fas clic dret i selecciones ”Restaurar arxiu”. T’apareixerà una finestra

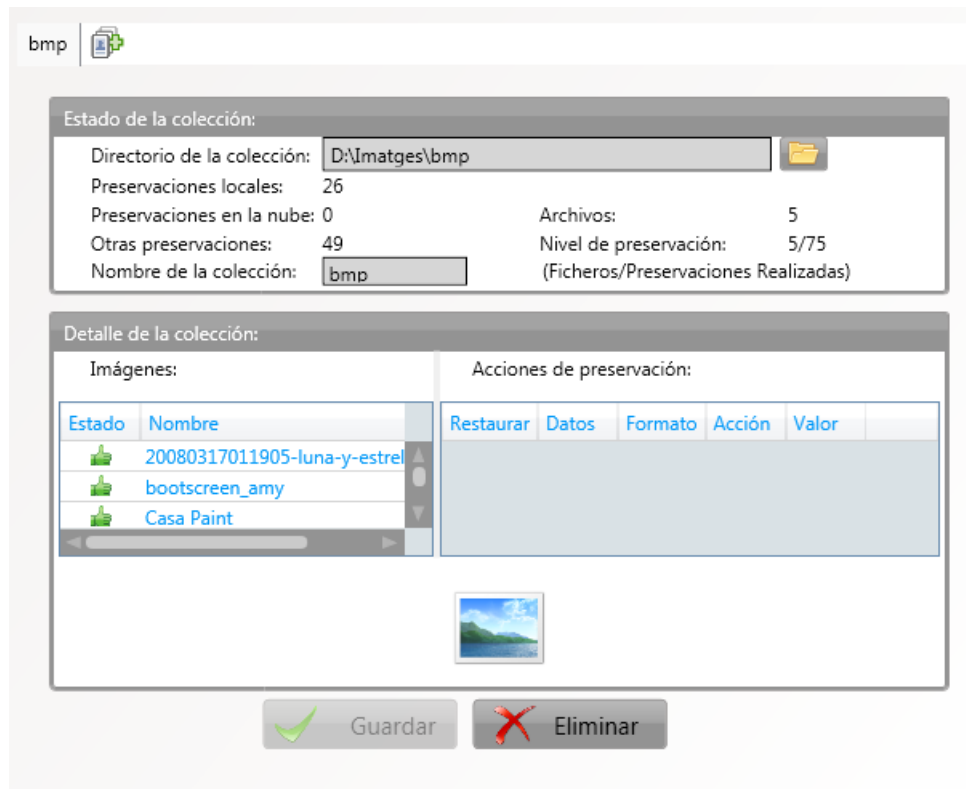


Figura 8.11: Pantalla àlbums.

per a valorar la preservació feta, la valores i tanques, ja has recuperat el teu arxiu.

#### 8.2.3.4 Amics

L'aplicació ens permetrà gestionar també una llista d'amics:

- **Afegir:** Per afegir un amic, s'ha de fer clic al botó "Afegir" on s'haurà d'entrar el correu-e de l'amic i quina capacitat de disc donarem per a que pugui preservar els seus arxius.
- **Eliminar:** Selecciones l'amic de la llista, cliques en el botó eliminar i ja te n'has desfet.
- **Modificar:** Pots canviar l'espai cedit a l'amic modificant el valor del camp d'"espai atorgat".

- **Consultar:** En clicar un amic de la llista, pots consultar informació com el nick, el correu, l'espai que li atorgues i els formats de fitxers que te.

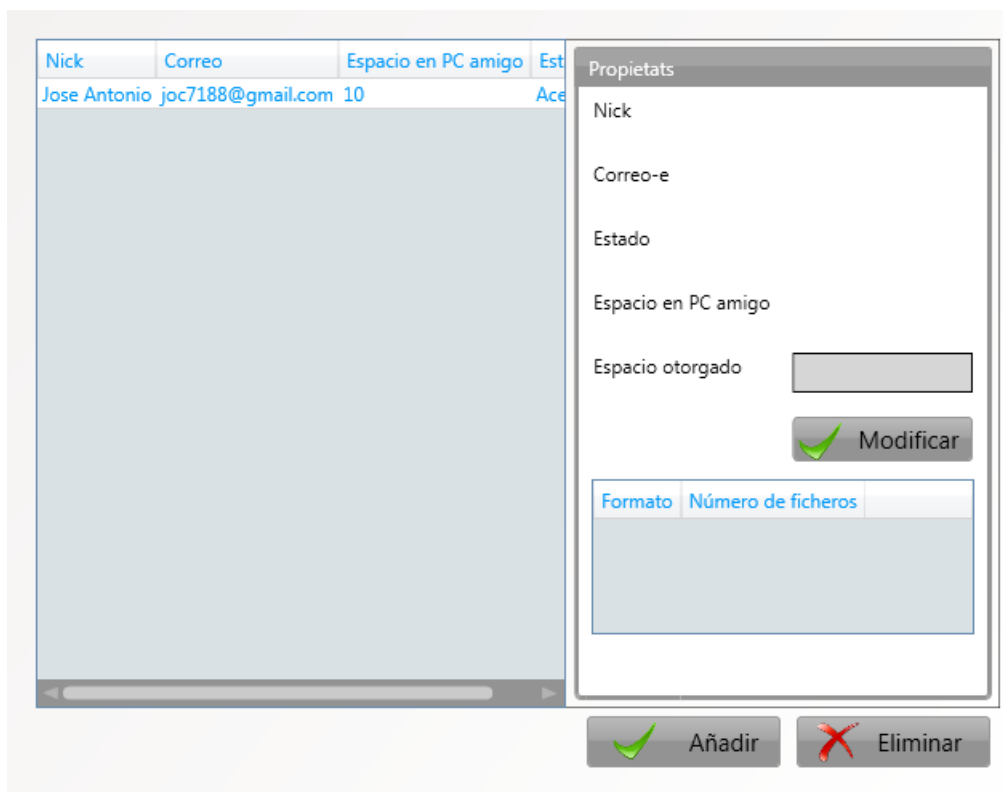


Figura 8.12: Pantalla amics.

# Annex A

## It is Forever

El motor de preservació de Pyramid ha servit per a fer una aplicació web anomenada It Is forever[36].



Figura A.1: It Is Forever Logo.

### A.1 Introducció

L' **It Is Forever** és un servei exclusiu que té com a objectiu regalar els moments més especials copsats en una fotografia. El regal pot ser virtual o físic, es pot personalitzar segons l'ocasió, i la foto i els continguts seran preservats per sempre. La versió virtual o online permet a l'usuari triar el motiu del regal a la xarxa i adjuntar la fotografia a regalar. A partir d'aquí, es crea un espai web personalitzat que conté la imatge preservada, un missatge personalitzat, una carta de benvinguda amb una clau d'accés per accedir a aquest espai i un certificat de garantia de preservació. A més, aquest espai web es pot compartir en diferents xarxes socials. La segona opció és el físic: si l'usuari ha triat regalar el paquet físic, a més de tenir les mateixes

prestacions del regal online, rebrà el regal a casa seva. Dins d'aquest paquet hi trobaran la fotografia, la documentació d'accés a la versió en xarxa del regal i la garantia de preservació.

## A.2 Què és?

It Is Forever és una plataforma web que vol satisfer les necessitats de preservació digital de les fotografies dels teus millors moments.

Tots volem atrapar els millors instants de les nostres vides en una fotografia i volem que aquesta quedi guardada per sempre. Amb It Is Forever podràs preservar tots aquests moments i els podràs regalar a les persones que més estimes.

## A.3 Unió a la inversa

Finalment, es va fer una unió a la inversa, ja que el motor de Pyramid, preserva les imatges a It Is Forever, es va pensar en que els usuaris de It Is Forever poguessin gestionar els seus regals a través de Pyramid.

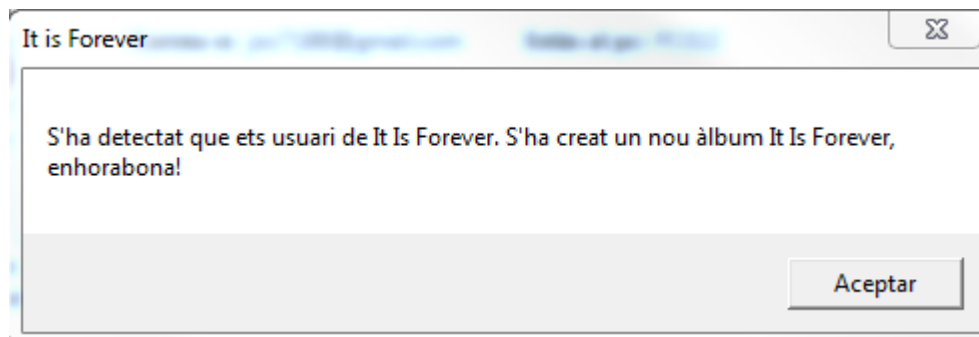


Figura A.2: Nou ItIsForever.

Si l'usuari de It Is Forever es fa un compte a Pyramid amb el mateix correu electrònic, Pyramid veurà que hi ha aquest vincle i donarà un avís conforme ha detectat aquest vincle i crearà un àlbum ItIsForever amb les imatges que tens a l'It Is Forever.

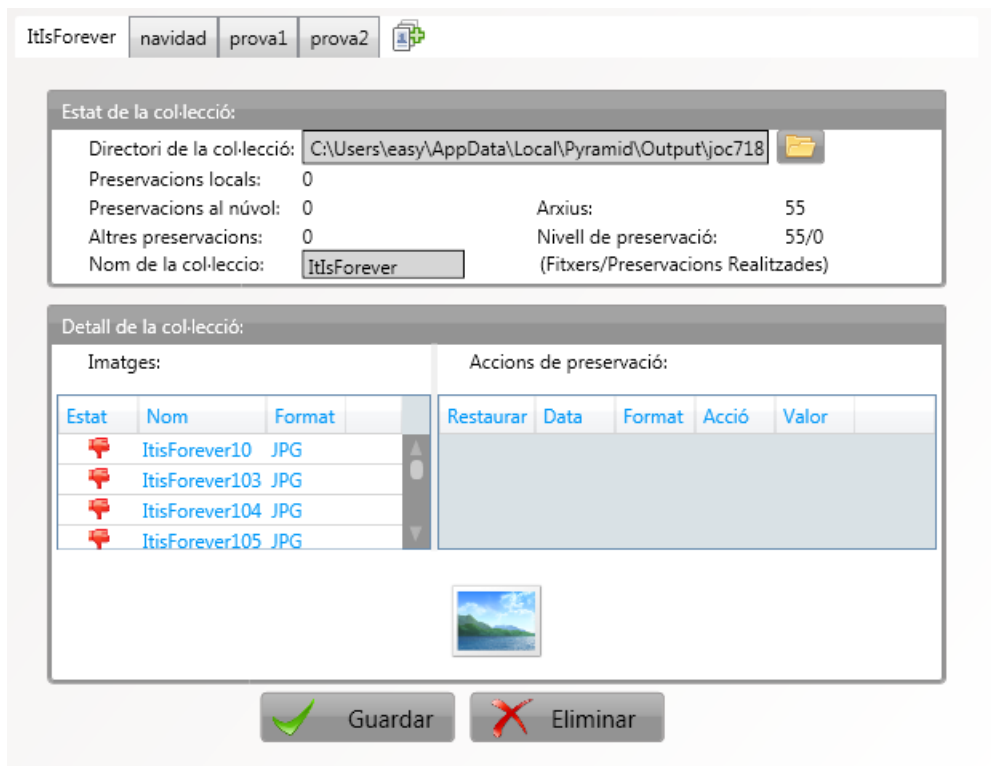


Figura A.3: It Is Forever Àlbum.



Finalment, es va haver d'afegir una taula a la base de dades per a tenir constància dels regals que s'han obtingut.

Diagrama entitat-relació

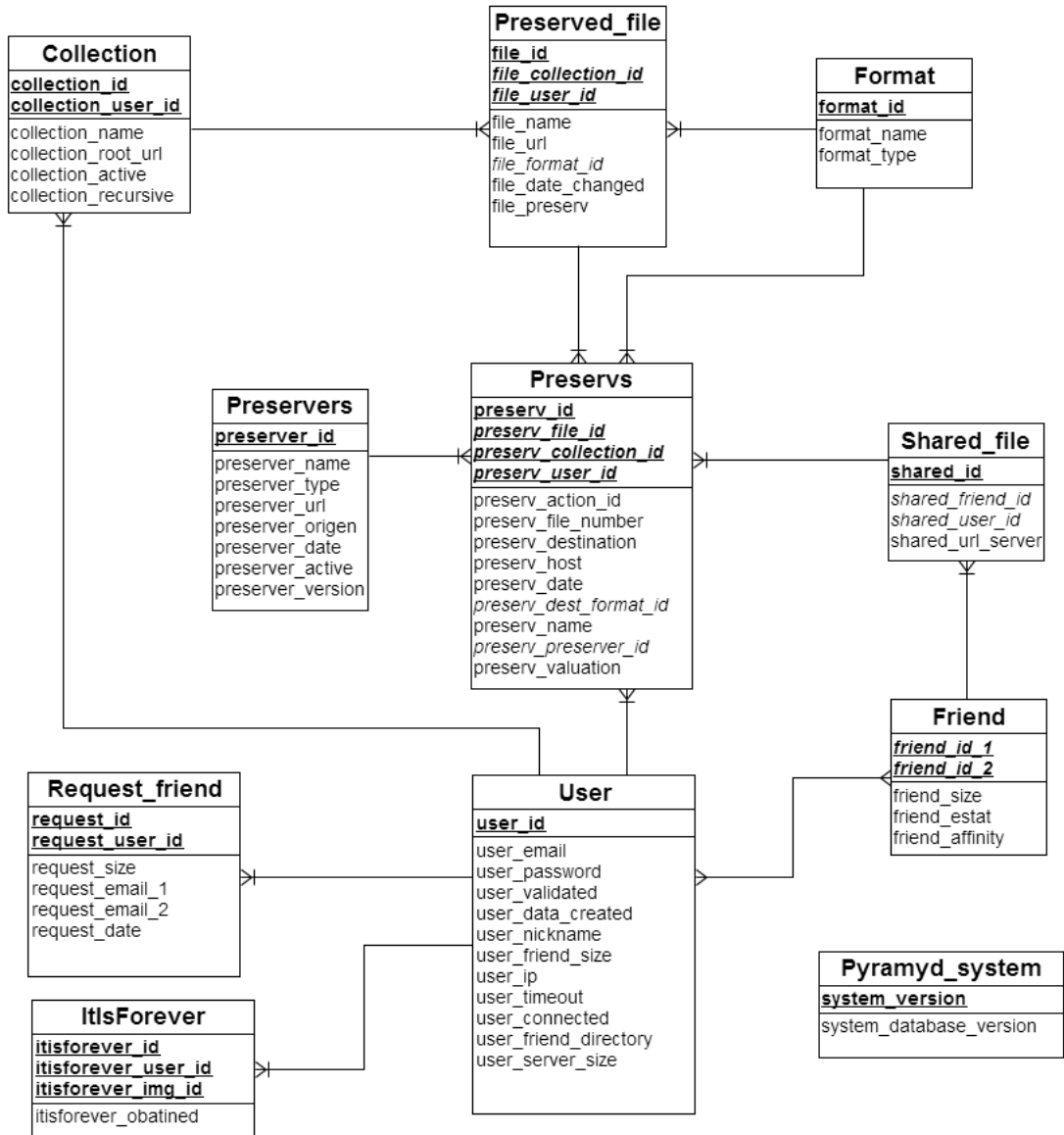


Figura A.4: Entitat relació final.

## Annex B

# Problemes en el Desenvolupament Basat en Agents

Segons Wooldridge [42], alguns dels problemes que apareixen en desenvolupar una aplicació basada en agents són els següents:

- No entendre on els agents poden ser útils o sobreestimar-los: tot i que els agents representen una nova manera i important de desenvolupar software, té les seves limitacions i és necessari conèixer-les. Els agents, segueixen essent software i com a tals estan regits per les mateixes limitacions. A l'hora, les seves dots d'intel·ligència estan lligades amb l'estat de l'art de la Intel·ligència Artificial.
- Ser dogmàtic amb els agents: malgrat que es puguin aplicar en un ventall ampli d'aplicacions, els SMA no són una solució universal i en alguns casos altres paradigmes són més apropiats. Si a més a més, hi afegim la immaduresa relativa d'aquesta tecnologia i les poques aplicacions d'agents que existeixen, a l'hora de desenvolupar una aplicació mitjançant agents cal tenir molt clar les avantatges que oferirà.
- No saber perquè es vol fer ús dels agents: En molts projectes es volen aplicar els agents sense saber què ofereixen o quin paper resoldran en ell.
- Voler crear una solució massa genèrica: Un dels problemes més freqüents és voler crear una aplicació massa genèrica quan el que es necessita n'és una dissenyada a mida.
- Creure que els agents són una bala de plata. Com a bala de plata s'entén una tècnica que permetrà una millora en l'ordre de magnitud en el desenvolupament del software. La tecnologia agent encara és un paradigma molt recent i

encara no està provat de manera suficient, tot i això és probable que algú la proclami una bala de plata.

- Oblidar-se que s'està desenvolupant software. Malgrat la gran quantitat de metodologies existents per dissenyar i desenvolupar aplicacions amb agents, aquestes no són prou madures. Molts desenvolupadors obliden les bones pràctiques del disseny software.
- Oblidar que s'està dissenyant software amb múltiples fils d'execució. Els sistemes amb múltiples fils d'execució han sigut reconeguts com uns dels sistemes més complexos tant de dissenyar com d'implementar. Els sistemes multiagent tendeixen a tenir-ne múltiples, cosa que cal no perdre de vista.
- El disseny no explota la concurrència. Quan en un disseny la concurrència és mínima, cal questionar-se realment si aquest paradigma resulta útil.
- Decidir que es vol una arquitectura d'agents pròpia. Les arquitectures són plantilles per construir agents, construir-ne una de nova per a cada aplicació no és una opció recomanable.
- Els agents utilitzen massa IA. Centrar-se només amb la intel·ligència, sovint amb tècniques experimentals, pot sobrecarregar-los i implicar que deixin de ser usables.
- Veure agents a tot arreu. Quan s'aprèn què són els agents, hi ha la tendència en veure'ls a tot arreu. Tot i que es podria aplicar, a la solució li mancava eficiència.
- Tenir massa pocs agents. De la mateixa manera que amb la programació orientada a objectes no s'opta per implementar tota la funcionalitat en una sola classe, tampoc s'ha de fer en aquest paradigma.
- Gastar tot el temps implementant la infraestructura. Atès que no hi ha plataformes massa utilitzades pel disseny de sistemes multiagent, en molts projectes s'acaba dedicant una gran quantitat de temps per tal d'implementar llibreries i eines software, que al cap i a la fi fan poc més que intercanviar missatges a través de la xarxa.
- Els agents interactuen d'una manera desorganitzada. El dinamisme dels sistemes multiagent són complexos. Un dels altres errors de concepte, és que els sistemes basats en agents no necessiten realment una estructura. Sovint cal estructurar la societat d'agents per tal de reduir-ne la complexitat alhora que se n'incrementa l'eficiència.

# Bibliografía

- [1] Pyramid. <http://www.pyramid.cat/>
- [2] Projecte *PROTAGE*. <http://www.ra.ee/protage>
- [3] *ICONIX* <http://iconix-software.blogspot.com.es/>
- [4] ClockingIT <http://www.clockingit.com/>
- [5] Russell, S. J. and Norvig, P. (2002). Artificial Intelligence: A Modern Approach (2nd Edition). Prentice Hall.
- [6] A. Newell. The knowledge level. *Artificial Intelligence*, 18:87-127, 1982.
- [7] M. d’Inverno and M. Luck. Understanding agent systems. Springer series on agent technology. 2001.
- [8] J. Ferber. Multi-Agent Systems. Addison Wesley, 1999.
- [9] S. Franklin and A. Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, volume 1193, 1996.
- [10] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115-152, 1995.
- [11] Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons, Inc., New York, NY, USA.
- [12] J. M. Corchado and J. M. Molina. *Introducción a la Teoría de Agentes y Sistemas Multiagente*. Edite Publicaciones Científicas, 2002.
- [13] K. P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79-92, 1998.
- [14] Gladney, H. M. (2006). Principles for digital preservation. *Commun. ACM*, 49(2):111-116.

- [15] Strodl, S., Becker, C., Neumayer, R., and Rauber, A. (2007). How to choose a digital preservation strategy: Evaluating a preservation planning procedure. In Proceedings of the 7th ACM IEEE Joint Conference on Digital Libraries (JCDL'07), pages 29-38, New York, NY, USA. ACM Press.
- [16] Garrett, J. R. (1995). Task force on archiving of digital information. Technical report.
- [17] Waugh, A. (2000). Preserving digital information forever. In Proceedings of the conference ACM Digital Libraries, pages 175-184. San Antonio.
- [18] Rothenberg, J. (1995). Ensuring the Longevity of Digital Documents. *Sci. Amer.*
- [19] Granger, S. (2000). Emulation as a digital preservation strategy. *D-Lib Mag.* 6 (10). <http://www.dlib.org/dlib/october00/granger/10granger.html>.
- [20] Russell, K. (2000). Digital preservation and the cedars project experience. In Proceedings of the International conference on Preservation and Long Term Accessibility of Digital Materials, pages 139-154. York. England.
- [21] Woodyard, D. (2000). Digital preservation: The Australian experience. *National Library of Australia Staff Papers*, 0(0)
- [22] Gilheany, S. (1998). Preserving information forever and a call for the emulators. In Proceedings Digital Libraries Conference and Exhibition: The Digital Era: Implications, Challenges and Issues. Singapore.
- [23] Rothenberg, F. (1999). Avoiding technological quicksand: Finding a viable technical foundation for digital preservation. A Report to the Council on Library and Information Resources, Council on Library and Information Resources, Washington, DC.
- [24] Lee, K., Slattery, O., Lu, R., Tang, X., and McCrary, V. (2002). The state of the art and practice in digital preservation. *Journal of Research of the National Institute of Standards and Technology*, (107):93-106.
- [25] Hendley, T. (1998). Comparison of methods and costs of digital preservation. *British Library Research and Innovation Report*, (106):121.
- [26] Muñoz, A. M., López, J. A. & Caicedo, E. F., Inteligencia de enjambres: sociedades para la solución de problemas (una revisión). <http://www.revistas.unal.edu.co/index.php/ingevin/article/viewFile/14901/15706>
- [27] T. White. Swarm Intelligence, <http://www.sce.carleton.ca/netmanage/tony/swarm.html>, 2007

- [28] Vitorino Ramos, Carlos Fernandes, Agostinho, C. Rosa Social Cognitive Maps, Swarm Perception and Distributed Search on Dynamic Landscapes, Brains, Minds & Media, Journal of New Media in Neural and Cognitive Science, NRW, Germany, 2005, <http://arxiv.org/abs/nlin/0502057v1> [nlin.AO]
- [29] Dorigo, M., Maniezzo, V., Colorni, A., Ant system: An autocatalytic optimizing process., Technical Report 91-016, Politecnico di Milano, Italy, 1991a.
- [30] Bernardo A. Huberman, Tad Hogg. The Emergence of Computational Ecologies 1992 Lectures in Complex Systems, pp. 185-205, Addison-Wesley 1993.
- [31] Wikipedia. [http://es.wikipedia.org/wiki/Computacion\\_distribuida](http://es.wikipedia.org/wiki/Computacion_distribuida).
- [32] Bernardo A. Huberman and Tad Hogg. The behavior of computational ecologies. In B. A. Huberman, editor, The Ecology of Computation, pages 77-115. North-Holland, Amsterdam, 1988.
- [33] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison- Wesley, NY, 1989.
- [34] D.E. Goldberg (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA.
- [35] C. Darwin (1859). On the Origin of Species by Means of Natural Selection, Murray, London
- [36] It Is Forever. <http://www.itisforever.com/>
- [37] M. Haynie, "Enterprise cloud services: Deriving business value from Cloud Computing," Micro Focus, Tech. Rep., 2009.
- [38] Gerard Briscoe, Alexandros Marinos. "Digital Ecosystems in the Clouds: Towards Community Cloud Computing"(2009 3rd IEEE International Conference on Digital Ecosystems and Technologies)
- [39] INGENIAS. 2005. Universidad Complutense de Madrid (GRASIA). <http://grasia.fdi.ucm.es/ingenias>
- [40] Caire, G., Leal, F., Chainho, P., Evans, R., Garijo, F., Gomez-Sanz, J. J., Pavon, J., Kerney, P., Stark, J., and Massonet, P.:Eurescom P907: MES-SAGE - Methodology for Engineering Systems of Software Agents. <http://www.eurescom.de/~publicwebpace/P900-series/P907/index.htm>

- [41] de la Rosa, J.L., Trias, A., Ruusalepp, R., Aas, K., Moreno, A., Roura, E., Bres, A., and Bosch, T. (2010). Agents for social search in long-term digital preservation. In Accepted by the 2010 International Conference on Semantics, Knowledge and Grids (SKG 2010).
- [42] Wooldridge, M. (2002). An Introduction to MultiAgent Systems. John Wiley & Sons, Inc., New York, NY, USA.
- [43] de la Rosa, J. Ll. and Olvera, J.A. 2012. First Studies on Self-Preserving Digital Objects. Artificial Intelligence Research and Development - Proceedings of the 15th International Conference of the Catalan Association for Artificial Intelligence, CCIA 2012, Volume 248, pp: 213 - 222, 2012, Novembre, Alacant, Spain.
- [44] José Antonio Olvera and Josep Lluís de la Rosa, An Outline of the Application of Agents to Digital Preservation and an Introduction to Self Preservation Aware Digital Objects, 13th European Agents Systems Summer School, Girona, Catalonia, July 1-15, 2011, online access: <http://eia.udg.edu/easss2011/resources/docs/paper8.pdf>
- [45] Josep Lluís de la Rosa, Albert Trias, Antoni Martorano, Eloi Colomeda, David Huerva, and Esteve del Acebo, Shout and Act, ISSN 0922-6389, Artificial Intelligence Research and Development, Vol. 202 pp: 91- 100 , Nov 2009, IOS Press, ISBN: 978-1-60750-061-2
- [46] J.L. de la Rosa, Albert Trias, Esteve del Acebo, Silvana Aciar, and Hugo Quisbert, Shout and Act: an Algorithm for Digital Objects Preservation inspired from Rescue Robots, In DP 1st International Workshop in Digital Preservation, JCDL (A+ congress), June 19, 2009, Austin, Texas, USA
- [47] Lugar de publicación: Amsterdam [de la Rosa et al., 2010] Josep Lluís de la Rosa, Albert Trias, Raivo Ruusalepp, Kuldar Aas, Alex Moreno, Eloy Roura, Albert Bres, and Teresa Bosch, Agents that Supply Knowledge exchange in Long-Term Digital Preservation, to appear in the eChallenges 2010 Conference, Warsaw, October 27-29, 2010.
- [48] [de la Rosa et al., 2008] Josep Lluís de la Rosa, Johan E. Bengtsson, Raivo Ruusalepp, Ann Hägerfors, and Hugo Quisbert, Using Agents for Long-Term Digital Reservation: the PROTAGE Project, Series Advances in Soft Computing Publisher - International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), Vol. 50/2009 pp:118-122, ISSN 1615-3871 (Print) 1860-0794 (Online, Springer Berlin / Heidelberg, DOI 10.1007/978-3-540-85863-853)

- [49] [Jin et al., 2010] Xiaolong Jin, Jianmin Jiang, and Josep Lluís de la Rosa, PROTAGE: Long-Term Digital Preservation Based on Intelligent Agents and Web Services, ERCIM News, no. 80, January 2010.
- [50] [de la Rosa and Olvera, 2012] JL de la Rosa and JA Olvera, Studies on Self-Preserving Digital Objects, accepted in CCIA2012 and to be published in IOS press, October 2012
- [51] [de la Rosa et al., 2011] Josep Lluís de la Rosa, Nicolás Hormazábal, Silvana Aciar, Gabriel Lopardo, Albert Trias, and Miquel Montaner, A Negotiation Style Recommender Based on Computational Ecology in Open Negotiation Environments ISSN: 0278-0046, IEEE Transactions on Industrial Electronics, Vol. 58 (6) : 2073-2085, June 2011
- [52] [Jose Antonio Olvera Cañizares, 2011] Anàlisi, disseny i prototipatge de nous paradigmes de preservació digital.